Semantic Web Service Process Mediation in WSMO:

Current Solutions and Open Issues

Kanmani Munusamy, Mohd Sapiyan Baba
Faculty of Computer Science & Information Technology,
University Malaya (UM),
Kuala Lumpur, Malaysia
{kanmani, pian}@um.edu.my

Suhaimi Ibrahim, Harihodin Selamat, Keyvan Mohebbi, Mojtaba Khezrian
Advanced Informatics School (AIS),
Universiti Teknologi Malaysia (UTM),
Kuala Lumpur, Malaysia
{suhaimiibrahim@, harihodin@, mkeyvan2@live, kmojtaba3@live}.utm.my

***Abstract*—Process mediation plays an important role in ensuring successful interaction between a provider and a service requestor. Therefore process mediation could be conceptualized as a 'middleware' that coordinates the interaction between web services. The Semantic Web Service promises automation in discovery, selection and composition but is still facing serious challenges in resolving mismatches where the Web service interaction takes place. For this paper, the WSMO, a Semantic Web Services framework is chosen and the current process mediation approaches that have adopted this framework are analyzed. The findings enable the identification of some open issues and process mediation elements. These identified factors can be further explored to support automatic communication mismatches in the generic Web Services.**

Keywords-Semantic Web Service; Process Mediation; Communication Mismatches; Mismatch Patterns; Choreography Interface

I. INTRODUCTION

Web service is one of the rapidly growing technologies that have been widely adopted by many organizations in industry. The main goal of the web service is to produce software component and business application that are available via the standardized interfaces. As there is an extensive increase in the number of Web Services, the needs of automation for discovery, selection and composition of these Web Services have risen. In order to bring an automation task into a web service, a semantic description on the method of invoking a service, the way the service works, the order of calling a service and the functionalities it offers has to be added to the Web Services. There are two well-known Semantic Web Services Frameworks and these are the OWL-based web service (OWL-S) [1] and Web Service Modeling Ontology (WSMO) [2].

Many Semantic Web Services research works are focused on automation of discovery, selection and composition of the Web Services which are aided by ontology. Research findings have highlighted that the most challenging tasks during automatic discovery, selection and composition of the Semantic Web Services are diagnosing and resolving incompatibility between Web Services. As a result an important terminology "Mediation" in Semantic Web Services has emerged to handle incompatibility between Web Services.

Fensel and Bussler [3] have described mediation as "a *process for settling a dispute between two parties where a third one is employed whose task is try to find common ground that will resolve inconsistencies between their respective conceptualizations of a given domain*". Apart from the definitions of Fensel and Bussler, there are many other definitions for mediator in context of Web Services. For instance, Grahne and Kiricenko [4] define mediator as a "*software module that provides sharing of services and agglomeration of resources into complex service"*.

There are three types of mediation, namely data, functional and process mediation. There are a significant number of researches on the Semantic Web Services that have explored data and functional mediation which is essential for automatic discovery, selection and composition. On the other hand process mediation has only been introduced as a supporting component in the composition of Web Services.

This paper focuses on process mediation in WSMO. For this study, current solutions are explored and open issues that are needed to be addressed and identified to support process mediation. It is clear that data mediation is an important element in process mediation and there is the dire need to mediate each incoming and outgoing data, before understanding the interaction between them. There are many existing researches on data mediation that support process mediation [5, 6] and therefore, the semantic or ontology in process mediation approaches are not mentioned in this paper.

The techniques identified have been used in understanding the interaction between the Web Services. It has been found that the existing process mediation approaches in the WSMO framework are tailored to a specific web service interaction scenario. There are many

 ISBN: 978-1-61208-152-6

elements needing to be explored to support the automatic generic Web Service solutions.

Here on this paper is organized as follows: Section 2 describes the process mediation in WSMO Framework and provides definitions of process mediation and mediator components. It also explains how choreography interface supports process mediation. Section 3, describes process mediation approaches that uses the WSMO Framework. Section 4, provides a discussion on the current approaches and addresses the open issues that need to be explored to generate process mediator automatically for generic solutions. Finally, Section 5 provides some discussion and the conclusions.

II. PROCESS MEDIATION IN WSMO

This section describes the role of the process mediator in resolving mismatches in messages. It explains each type of the message mismatches and the ways to resolve them. All the important component of WSMO that play important roles in process mediation have been summarized as follows.

A. Role of Process Mediator

The process mediator is called the communication mediator in WSMO. Fensel and Bussler [3] identified three types of communication mismatches between the Web Services, namely precise match, solvable and irresolvable mismatches. A precise match occurs when the sender web service sends the message in the exact order that the receiver web service has requested. Therefore it only requires data mediation to solve possible data or format mismatches. The unsolvable message mismatches usually comes to a dead end. This paper focuses on the solvable mismatches that have been highlighted by many researchers. There are five situations that generate solvable message mismatches as stated below:

1) sender Web Service sends a message that is not expected by the receiver
2) sender sends single message that is expected to be in multiple forms
3) sender sends multiple messages that are expected to be in the single form
4) sender sends messages in wrong order
5) sender is not sending messages that are expected by the receiver.

Cimpian [7] has presented five ways that the process mediator could address solvable mismatches as listed below:

1) it stops the original message since it is not requested by the receiver
2) it splits the original message before reaching the receiver
3) it combines the original message before it reaches the receiver
4) it inverses the original messages before it reaches the receiver
5) It sends a dummy message since a message is expected by the receiver

Similarly, there are many researchers [8, 9] who have identified interaction patterns that are able to transform an original message into the required communication pattern.

B. Mediator as WSMO Component

This framework provides a rich description of all the related aspects of Web Services through four important components which are: the goal, web service, ontology and the mediator. The ontology component plays an important role in this framework since it carries the semantic description for all the other components in this framework. The goal component defines the user's preferences with respect to the requested functionality and interfaces through the *requestedCapability* and the *requestedInterface*. On the other hand, the web service component defines the offered functionalities and the ways to interact with the services through the capability and interface elements.

Generally, the goal, the web service and the ontology components play a common role to bring the semantic description to a Web Service which is similar to other Semantic Web Services Frameworks. However, this framework has proposed a distinctive component known as the mediator to resolve the interoperability problems in Web Services at various levels such as data, functional and process mismatches. This component contains four elements which are the *OOMediator*, the *GGMediator*, the *WGMediator* and the *WWMediator* to overcome interoperability problems between different the WSMO components.

Based on the definition provided in [2], the *WGMediator* and the *WWMediator* are closely related to the process mediator. However, the implementation of the *WWMediator* in resolving these process mismatches is not specified clearly in any of the provided example.

C. Choreography Interface that Supports Process Mediation

This section describes how process mediation takes place in the WSMO framework. The process mediation is closely related to the interface element of the goal and the web service components. The interface element describes how the functionality of a service can be obtained from two perspectives namely the choreography and orchestration interface.

The choreography interface explains communication methods between the service provider and the requestor whereas orchestration interface explains the communication methods among several Web Services. This paper limits the process mediation in the choreography interface due to limited resources available for process mediation in the orchestration interface. The two main elements in the choreography interface that supports process mediation are state signatures and transition rules. Figure 1 illustrates the elements in the choreography class which is extracted from [10].

 ISBN: 978-1-61208-152-6

Class choreography
hasNonFunctionalProperties **type** nonFunctionalProperties
hasStateSignature **type** stateSignature
hasTransitionRules **type** transitionRules

Figure 1. Choreography Interface

In WSMO, the choreography interface is described as using state-based technique which is based on Abstract State Machine (ASM) methodology. They state signature plays important role to define the mode or state of each instances it is used in the choreography interface. This state signature elements are described by an ontology in WSMO. Below are the five modes of state signatures as described in [10]:-

1) Static: any instance of relation and concepts in the "static" mode cannot be changed by both the provider and requester's choreography interface.

2) In: any instance of relation and concepts in the "in" mode can only be read by the choreography interface. It also means that the instance in "in" mode is expected as input by the choreography interface to invoke the service.

3) Out: any instance of relation and concepts in the "out" mode can only be created by the choreography interface. It also means that the instance that in "out" mode will be produced as an output during the invocation.

4) Shared: any instance of relation and concepts in the "shared" mode can be read and created by both the choreography interface.

5) Controlled: any instance of relation and concepts in the "controlled" mode can only be created and modified by the choreography interface.

These state signatures do not return the actual value of the instances during the invocation. They only contain a Boolean value (true or false). It returns true when an instance is required by the corresponding service such as instance with "in" and "shared" mode. It will also be stored in the internal repository since it can be useful in communication between the services.

The second important element in the choreography interface is the transition rules. They also termed as guarded transitions. A rule is triggered when the current state of the instance fulfills certain conditions. The rule does not reflect the actual system processing of the instance value. However, it expresses the data flow between the interacting Web Services.

III. PROCESS MEDIATION APPROACHES USING WSMO FRAMEWORK

In this section, we will discuss four process mediation approaches that adopt the WSMO Framework and these are the message-based process mediation, the process mediation algorithm in Semantic Web Service (SWS) challenge, the process mediator as goal in IRS-III and the space based process mediation in Triple Space Computing (TCS).

A. *Message based process mediation*

In this WSMO Framework, the process mediation at runtime is as proposed in [11, 12]. It mediates communication mismatches between the provider and requestor by analyzing the state signature and transition rules in the goal and web service components.

Figure 2 illustrates process mediation in WSMO and the interaction between the choreography interfaces of the goal and web service. It also illustrates the role of the state signatures and transition rules in the process mediation and the main components in the process mediation such as the Choreography Parser, Internal Repository and the WSML Reasoner and Data Mediator.

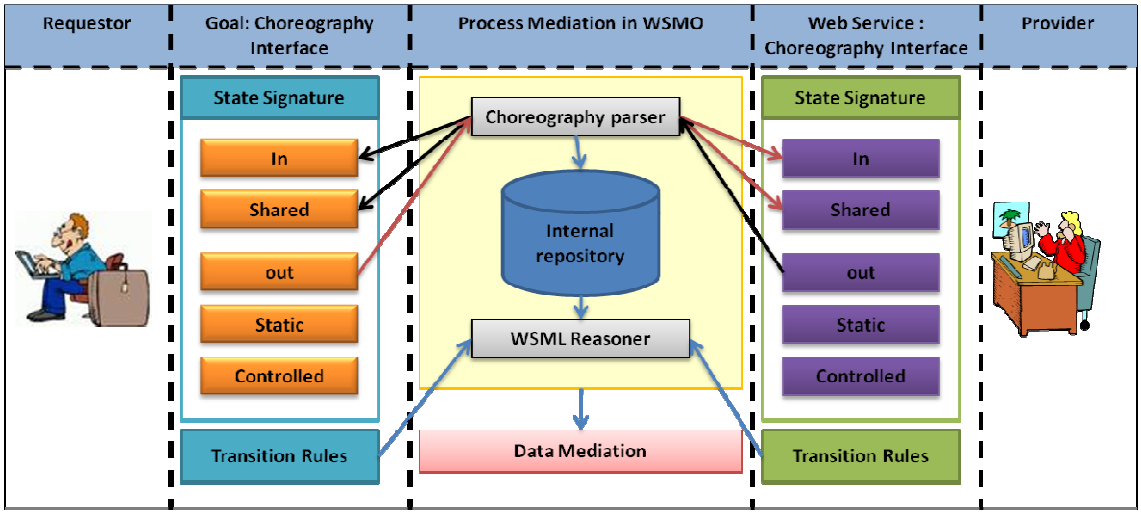

Figure 2. Message-based Process Mediation

This approach uses the Choreography Parser to check the mode of each state signature before it stores them in the Internal Repository. It matches the “in” mode instance in a goal/web service interface with the “out” mode instance from the web service/goal interface. This “in” and “out” list defines the data flow between the provider and the requestor. This data flow analysis is supported by the transition rule which defines the sequencing of the data exchange. This approach uses the WSML Reasoner to evaluate the transition rules before sending or deleting the stored instances.

However, this approach provides insufficient discussion on how choreography parser which matches the state signatures and the WSML Reasoner evaluates the transition rule that works together. Secondly, it also not utilizing the *WGMediator* or *WWMediator* components as explained in the WSMO concepts. Thirdly, it only provides steps to resolve process mismatches in a specific scenario but not for the general algorithm as to how the process mediator could perform the transformation based on the state signatures.

B. *Process mediation algorithm in SWS Challenge*

WSMO Framework has also extended to resolve the mediation scenario in the SWS challenge [13]. In this approach, the message interactions are evaluated using both the transition rules and the data flow that are extracted from the choreography interface. Two important contributions of this approach have been extracted in order to resolve the process mediation.

Firstly, it has defined four basic choreography rules that are derived from the WSDL operations. The WSDL operations are classified into four patterns; in-out, in-only, out-only, out-in based on the sequence of input or output messages of the operation. Secondly, this approach provides a general algorithm that handles the communication mismatches.

Generally, this algorithm collects all the data that needs to be exchanged between the Web Services and store them into the memory. Firstly, it evaluates the transition rules in each Web Services and stores required actions such as add and remove into the memory. At the same time, it also sends the input parameter in each web service as stated in the choreography interface.

Secondly, it evaluates the action list in the memory and deletes the removed action and the corresponding data from the data list in the memory. It then checks the output symbols at each web service. The output symbols that are equivalent to the messages in the add action will be inserted into data list in the memory and removes the corresponding add action from the action list. This algorithm ensures that the each web service memory contains the expected incoming messages. Finally, it calls the data mediation to mediate the data in both Web Service memories.

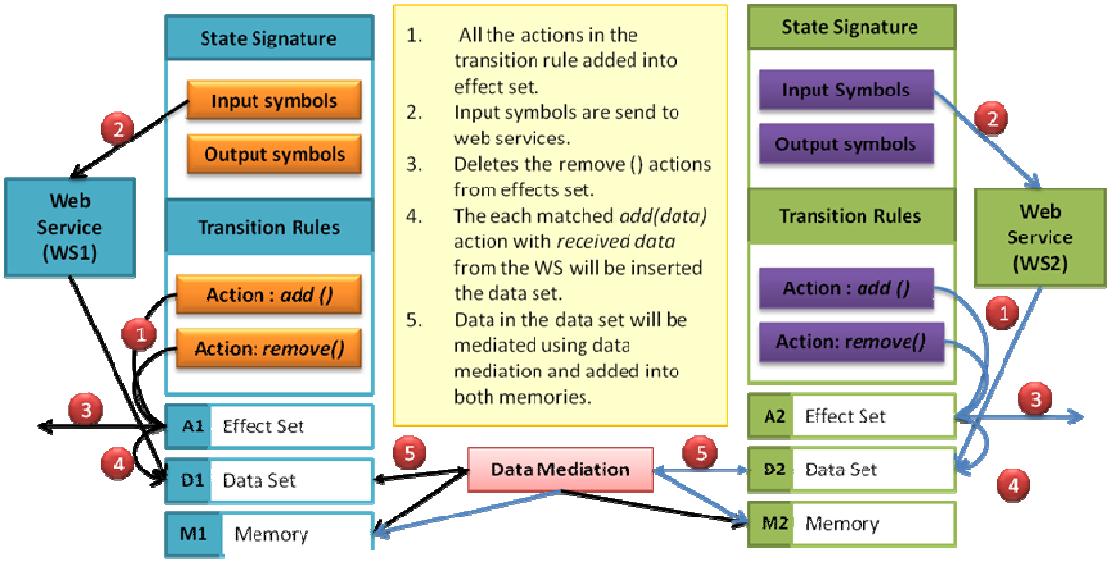

Figure 3. Process Mediation in SWS Challenge

Figure 3 illustrates how the algorithm inserts the mediated data into the memory of the Web Services based on the choreography interface. This algorithm has addressed all mismatch patterns [7] except for the new generating messages. It allows message ordering and stops unexpected messages. The message merging and spliting is handled by the data mediation. This approach has provided a clear view on how the transition rules and the state signatures can be evaluated using an algorithm to generate the process mediation steps. Generally, this process mediation technique is similar to space based process mediator (SPM) [14] approach. It has provided memory space to the each web service to handle the data flow which is supported by the choreography transition rules. It does not specify the usage of the WSMO mediator components such as the *WGMediator* or *WWMediator* in the process mediation algorithm. Moreover, ability of the algorithm in handling complex mismatches or combination

 ISBN: 978-1-61208-152-6

of more than one mismatch patterns are also underspecified.

C. Process mediator as goal in IRS-III

The main aim of Internet Reasoning Services (IRS) is to provide automated or semi-automated solutions for the semantically enhanced system over web. After few evolution of IRS, the IRS-III [15] has incorporated its existing framework which uses OCML, the ontology representation language with the WSMO core elements which are the goal, the web service and the mediator. This combination has produced a semantic broker-based approach that is able to mediate between the requester and the Web Services provider.

In IRS-III, the mediation task is resolved by the mediation handler component. This handler consists of the goal, the data and the process mediator. These mediators serve as a bridge between the semantic description such as the GG-Mediator, WW-Mediator, WG-Mediator and OO-Mediator with the other IRS components. The process mediator in this approach uses the GG and WW mediators to resolve four types of mismatches; a) not matched input/output, b) wrong order of input/output, c) output/input that needs to be split, and d) output/input that needs to be concatenated.

The process mediation in IRS-III is also closely related to the choreography and orchestration based service interactions. It also adopts ASM that contains states and transition rules to represent the interaction between the service provider and the requestor. However, it uses the forward-chaining-rule engine to execute the service interactions. In addition to the transition rules, this approach has defined choreography primitives to control the conversation between the IRS-III and Web Services.

Differing from WSMO/X approach, IRS-III does not load choreography interface of both goals of the requestor and web service provider. The IRS-III only evaluates the choreography interaction from the requester's perspective. Below are issues on process mediation in this approach.

- It has declares the mediators as goal which can be invoked as the mediation services. However, detailed explanation on how this mediation goal can be discovered and selected is based on the underspecified communication mismatches.
- The generation of the WW and GG Mediators supports the process mediator component is not specified.
- As for the other approaches, it also does not provide detailed description on reasoning mechanism used during evaluation on the transition rules.

D. Space based process mediation in Triple Space Computing

Apart from IRS-III, WSMO framework has also collaborated with Triple Space Computing (TSC) method to generate process mediation in the Semantic Web Services environment. TCS uses the Space-based Process Mediator (SPM) [14] approach to handle the communication mismatches between the service requestor and provider. The SPM method evaluates the data flow between the services using the data space. It also analyses the choreography rules that describe which data to be exchanged according to guarded rules through the control flow analysis.

TCS adopts the SPM method and provides a virtual data space, which is divided into the requestor and provider subspace. Generally, the TCS plays a middleware role between the requestor and provider, whereby all the sending and requesting messages take place through the TCS. It also handles process mediation by redirecting, transforming, stopping the data stored in the provider and requestor's sub space. The main feature of the TCS framework that supports the process mediation is the backend storage that provides shared virtual data space and the storage management mechanism that is able to store the history of interaction, monitor the interaction and resume interactions from point of failure.

In this approach [16], it has been described as to how the five resolvable message mismatches (as stated in WSMO) can be overcome by storing and transforming via the TCS virtual data space. However, the implementation of the actual WSMO concepts such as the goal, the web service, the mediator and the ontology in the TCS framework are underspecified.

IV. DISCUSSION

The main aim of this paper is to identify the important component of process mediation based on the existing approaches that are related to the WSMO framework. The existing techniques can be discussed by two main perspectives; namely, identification of the mismatches and generation of the process mediator.

Firstly, for the identification of the communication mismatches, all the approaches are uses data analysis together with process flow analysis. The data flow is analyzed by comparing the expected messages of a web service with the actual incoming messages and outgoing messages with the expected messages in the web service of the recipient.

Generally, the data flow analysis techniques are supported by the transition rules which describe how the Web Services interact with each other. However the detailed description of the reasoning mechanism that is applied and the usage of ontology during the analysis of the transition rules are not available. In [13], data mediation is presented after identifying the data that is needed to be stored in the repository based on the choreography analysis in individual web services.

All these approaches however do not identify the five process mismatches specified in [3] according to the mismatch patterns. They only ensure that these mismatches are addressed in the approaches. Generally, a structured analysis on the data flow analysis which is supported by transition rules is still regarded as an open issue in order to address solution for generic process mediation.

 ISBN: 978-1-61208-152-6

Secondly, the techniques analyzed are involved in resolving the communication mismatches and that process mediators can be presented as web services [11, 12], algorithms [13], goals [17] and as virtual data space [16]. However, location, invocation and management issues of these process mediator solutions are also underspecified. The *WGMediator* and *WWMediator* elements that are related to the process mediation as stated in WSMO framework are not discussed in the actual implementation except for [17]. Figure 4 summarizes the open issues in semantic process mediation in the WSMO framework.

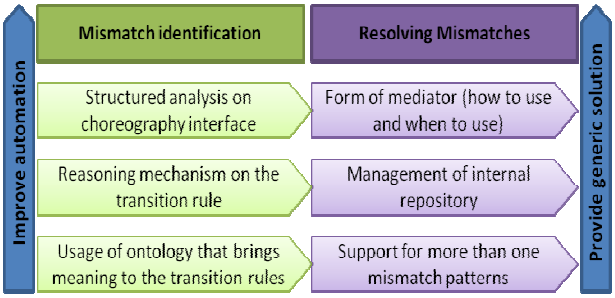

Figure 4. Open Issues in Process Mediation

V. CONCLUSION

For this paper, the important elements of the process mediation in the Semantic Web Services are based on the analysis of the approaches that are been collaborated with the WSMO framework. WSMO components that specifically support process mediation are discussed. This followed by the process mediation approaches that are related to the WSMO framework. Based on the analysis, the similarities and difference between each technique of process mediation is identified and the common elements that support process mediation have been extracted from reviewed literature for this paper. The findings reveal that these elements can be extended to support the automatic communication mismatches in the generic Web Services.

ACKNOWLEDGMENT

This research is supported by Ministry of Science and Technology and Innovation (MOSTI), Malaysia and Universiti Teknologi Malaysia (UTM) under Fundamental Research Grant Scheme (FRGS) Vote Number 4F054.

REFERENCES

[1] 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 (2004), "OWL-S: Semantic Markup for Web Services ", W3C Member Submission, Retrieved from: http://www.w3.org/Submission/OWL-S/, Last Access: 28 June 2011

[2] J.d. Bruijn, C. Bussler, J. Domingue, D. Fensel, M. Hepp, U. Keller, M. Kifer, J. Kopecky, H. Lausen, E. Oren, A. Polleres, D. Roman, J. Scicluna, and M. Stollberg (2005), "Web Service Modeling Ontology (WSMO)", W3C Member Submission, Retrieved from: http://www.w3.org/Submission/WSMO/, Last Access: 28 June 2011

[3] D. Fensel, and C. Bussler, "The web service modeling framework WSMF," *Electronic Commerce Research and Applications*, vol. 1, no. 2, 2002, pp. 113-137.

[4] G. Grahne, and V. Kiricenko (2005), "Process Mediation in Extended Roman Model", in M. Hepp, A. polleres, F. Harmelen, and M. Genesereth (Eds.) *First International Workshop on Mediation in Semantic Web Services (MEDIATE 2005) conjuction with 3rd International Conference on Service-Oriented Computing (ICSOC 2005)*, Amsterdam, Netherlands pp. 17-33.

[5] K. Gomadam, A. Ranabahu, Z. Wu, A.P. Sheth, and J. Miller, "A Declarative Approach using SAWSDL and Semantic Templates Towards Process Mediation," *Semantic Web Services Challenge*, 2009, pp. 101-118.

[6] Z.X. Wu, K. Gomadam, A. Ranabahu, A.P. Shetb, and J.A. Miller (2007), "Automatic composition of semantic web services using process mediation", in J. Cardoso, J. Cordeiro, and J. Filipe (Eds.), LSDIS lab, University of Georgia pp. 453-461.

[7] E. Cimpian, and A. Mocan (2005), "WSMX process mediation based on choreographies", *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, Nancy, pp. 130-143.

[8] X. Li, Y. Fan, S. Madnick, and Q.Z. Sheng, "A pattern-based approach to protocol mediation for web services composition," *Information and Software Technology*, vol. 52, no. 3, 2009, pp. 304-323.

[9] H.R. Motahari Nezhad, B. Benatallah, A. Martens, F. Curbera, and F. Casati (2007), "Semi-automated adaptation of service interactions", in *16th International World Wide Web Conference, WWW2007*, Banff, AB, pp. 993-1002.

[10] D. Fensel, H. Lausen, J.d. Bruijn, M. Stollberg, D. Roman, A. Polleres, and J. Domingue, "The Concepts of WSMO," *Enabling Semantic Web Services*, 2007, pp. 63-81.

[11] E. Cimpian, "Message-based Semantic Process Mediation," PhD Thesis, Faculty Science, National University of Ireland Galway, Ireland, 2010, p.198.

[12] E. Cimpian, and A. Mocan (2005), "Process Mediation in WSMX", in E. Cimpian (Ed.) Retrieved from: http://www.wsmo.org/TR/d13/d13.7/v0.1/, Last Access: 28 June 2011

[13] T. Vitvar, M. Zaremba, M. Moran, and A. Mocan, "Mediation using WSMO, WSML and WSMX," *Semantic Web Services Challenge*, 2009, pp. 31-49.

[14] Z. Zhou, S. Bhiri, W. Gaaloul, and M. Hauswirth (2008), "Developing process mediator for supporting mediated Web service interactions", in *Proceedings of the 6th IEEE European Conference on Web Services, ECOWS'08*, Dublin, pp. 155-164.

[15] L. Cabral, J. Domingue, S. Galizia, A. Gugliotta, V. Tanasescu, C. Pedrinaci, and B. Norton (2006), "IRS-III: A broker for semantic web services based applications", in Athens, GA, United states, Vol. 4273 LNCS, pp. 201-214.

[16] Z. Zhou, B. Sapkota, E. Cimpian, D. Foxvog, L. Vasiliu, M. Hauswirth, and P. Yu (2008), "Process mediation based on triple space computing", *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, Shenyang, pp. 672-683.

[17] J. Domingue, L. Cabral, S. Galizia, V. Tanasescu, A. Gugliotta, B. Norton, and C. Pedrinaci, "IRS-III: A broker-based approach to semantic Web services," *Web Semantics*, vol. 6, no. 2, 2008, pp. 109-132.

 ISBN: 978-1-61208-152-6