

SEMANTIC CONFLICTS DETECTION OF HETEROGENEOUS MESSAGES OF WEB SERVICES: CHALLENGES AND SOLUTION

Ibrahim Ahmed Al-Baltah, Abdul Azim Abdul Ghani,
Wan Nurhayati Wan Ab. Rahman and Rodziah Atan

Department of Software Engineering and Information System,
Faculty of Computer Science and Information Technology, Universiti Putra Malaysia, Serdang, Malaysia

Received 2013-12-30; Revised 2013-12-31; Accepted 2014-03-22

ABSTRACT

Service Oriented Architecture (SOA) is a new design style that has promised to alleviate the semantic interoperability problem between Web services. Exchanging the messages seamlessly between heterogeneous Web services is required but, it is still difficult due to data heterogeneity. As a result, a noticeable number of works has been proposed with the aim of solving this problem, but yet, the problem has not been solved efficiently. Furthermore, it is important to observe the lack of sufficient approaches in the state of the art, whose purposes are semantic conflicts detection between heterogeneous messages of Web services. It is for these reasons we take a step back to the detection step before providing the solution. This study proposes a new ontology-based approach that aim at detecting semantic conflicts between heterogeneous messages of Web services. The main purposes of this approach are to detect any conflict types between the messages during message exchange process and to identify the conflict type for each detected conflict. The proposed approach plays as a vital step for improving the semantic interoperability between heterogeneous Web service messages, since it facilitate the process of addressing semantic conflicts. A real scenario was used in order to demonstrate the feasibility of the proposed approach.

Keywords: Ontology, Conflicts Detection, Semantic Conflicts, Web Services, Data Heterogeneity

1. INTRODUCTION

Service Oriented Architecture (SOA) is a new design style for building software applications, in which the software components are loosely coupled (Sahin and Gumusay, 2008). Web services make the realization of SOA applications possible (Siew *et al.*, 2006). Accordingly, a huge number of Web services have been provided from different providers for different application domains.

From Web service implementation viewpoint, three main technologies that are commonly involved in Web service implementation, which are WSDL, SOAP and UDDI (Domingue *et al.*, 2008). WSDL is an XML-based description language for describing the Web

service interface elements such as types, messages, port type and binding (Christensen *et al.*, 2001). SOAP is an XML-based protocol for exchanging messages between Web services (Albreshne *et al.*, 2009). According to (Siluvai and Kumar, 2013), SOAP message consists of an envelope element, which contains optional elements include header and body element. SOAP envelop is a mandatory element that any message should have. SOAP header is an optional element for adding information to the message that would be used to SOAP intermediaries and the final destination of the message. SOAP body is a mandatory element that contains the actual message intended for the recipient of the message. UDDI is the common registry that supports the advertisement and discovery of Web services

Corresponding Author: Ibrahim Ahmed Al-Baltah, Department of Software Engineering and Information System,

Faculty of Computer Science and Information Technology, Universiti Putra Malaysia, Serdang, Malaysia

(Rathore and Suman, 2011). However, one of the most critical issues between Web services is data heterogeneity. This heterogeneity always prevents Web services to exchange their messages seamlessly. To be interoperable Web services, data heterogeneity should be addressed between their interoperating messages.

In practice, semantic interoperability is about ensuring that the precise meaning of the exchanged messages is preserved. In another word, semantic interoperability is concerned with ensuring that the Web services sender and receiver are communicating meaningfully. That is, the communicated Web services will share the same understanding about the data that being exchanged. In practice, semantic conflict always stands in the way of achieving semantic interoperability, so preventing the seamless exchanging of messages.

Establishing a meaningful communication requires both Web services sender and receiver to have the same interpretation and understanding of the data that being exchanged. Even though, SOA has promised to alleviate the semantic interoperability between services (Barnickel and Fluegge, 2010), semantic interoperability is still the crucial problem between Web services, which need to be addressed in order to exchange the messages seamlessly.

In fact, exchange the messages in a seamless manner is desirable, but unfortunately, in some cases, it is very hard to establish a seamless message exchange between heterogeneous Web services. The dominating reason for that is the existence of semantic conflicts between Web service messages. Semantic conflicts arise due to the use of multiple interpretations and representations of the same term and due to the use of multiple structures of the same application domain.

According to (Al-Baltah and Ghani, 2012), the process of tackling semantic conflicts between Web services messages revolves around three main steps; semantic conflicts identification, detection and solution. The identification step concerns with identifying the conflicts between Web services messages. The detection step is the process of detecting the conflicts between Web services messages. The solution step is the process of reconciling the semantic conflicts. However, the aim of this study is on detecting semantic conflicts between heterogeneous Web services at message level.

In this study, we propose a new ontology-based approach that aims at detecting and identifying the type of the conflict that being detected between the messages of Web services. Our detection approach plays as an aid to reconcile semantic conflicts; this is due to the fact

that, detecting and identifying the types of the detected conflicts is highly required in order to achieve semantic interoperability (Li and Ling, 2004).

The reminder of this study is organized as follows. Section 2 demonstrates the need for proposing a new approach to detect semantic conflicts at the message level of Web services. The challenges and the solution of semantic conflicts detection are represented in section 3. The researchers survey and discuss the related work in section 4. Section 5 introduces the proposed approach. The discussion is presented in section 6. Section 7 concludes the study with an outlook to future work.

2. THE NEED FOR SEMANTIC CONFLICTS DETECTION

In order to demonstrate the need for semantic conflicts detection, we present a real scenario which is reported in (Nagarajan and Verma, 2007). Let's consider the process of sending mails to customers of a company by using their phone numbers. Two Web services from different providers are participating in this scenario to get the full mailing address for the customer.

The first Web service is called GeoPhone, which is provided by service object provider. This Web service provides contact information based on the given phone number. The interface of this Web service is available at "<http://trial.serviceobjects.com/gp/GeoPhone.asmx?WSDL>". The second Web service is called "USGeocoding5" which is provided from strikeiron; the interface of this Web service can be accessed directly through "<http://ws.strikeiron.com/USGeocoding5?WSDL>". This Web service provides the demographic and logistical information for the provided address. This scenario starts by sending a request to GeoPhone service to provide the contact information of a specific phone number. The response (output) of GeoPhone service will be used as an input to USGeocoding5 to get the demographic and logistical information that is required from the company to send the mails. Exchanging the messages between these Web services is difficult due to the semantic conflicts between the output message of GetContactInfo (from GeoPhone service) and the input message of GetGeocodeUSA (from USGeocoding 5 service). **Figure 1** demonstrates the process of this scenario.

The red solid arrows in **Fig. 1** associate the elements of the output message with the corresponding elements of the input message, also they illustrate where the possible semantic conflicts may occur between the participated messages.

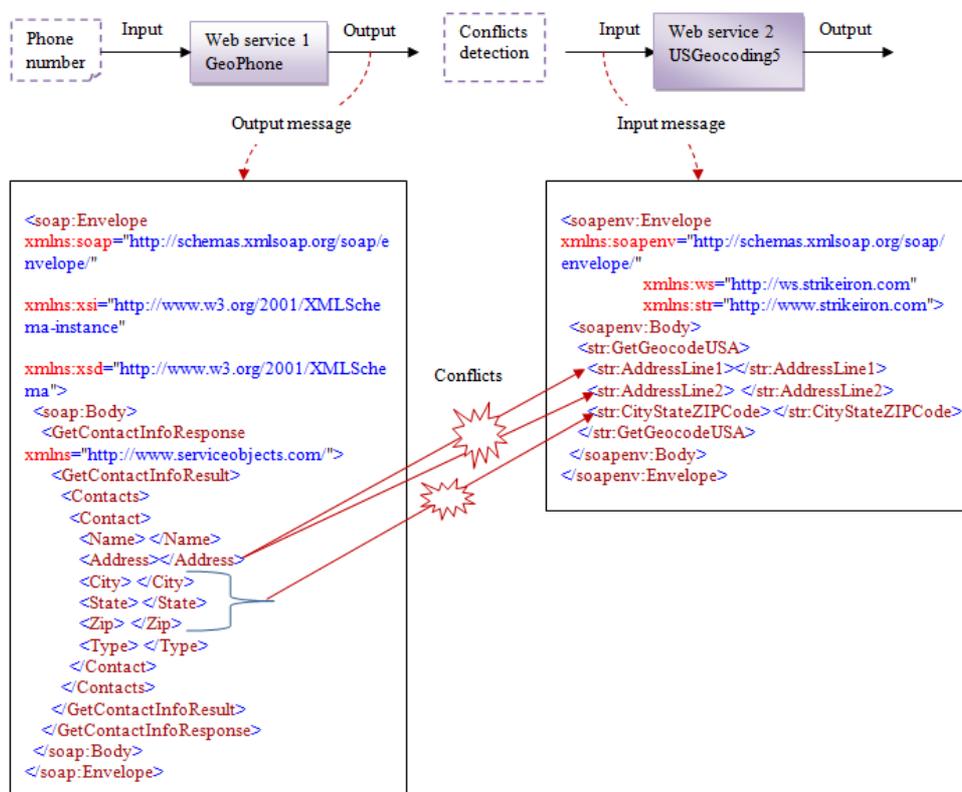


Fig. 1. Scenario illustrating the needs of semantic conflicts detection between heterogeneous Web services at message level

3. SEMANTIC CONFLICTS DETECTION: CHALLENGES AND SOLUTION

3.1. Challenges

In this study, the challenges of semantic conflicts detection were categorized into three main challenges semantic interpretation, representation and message structure.

Semantic interpretation: Generally, semantic interpretation refers to the mapping process between the syntactical term and its meaning (Hirst, 1992). To make this process successful, for every term there has to be one and only one interpretation of its meaning at least within the same context. From the practice point of view, a term can be subject to different interpretations in different contexts (Xu and Cheung, 2004) and this can only escalate the challenge of semantic conflicts detection. This is because multiple interpretations of the same term will trigger semantic conflicts. An approach that is dedicated to detecting these conflicts should map the term to only one interpretation of its meaning.

In any Web service domain, it is this challenge which makes the process of semantic conflicts detection to be difficult. This consequence follows from the fact that, Web services are distributed on the Web, which is an open, dynamic and distributed environment (Blake and Gomaa, 2005). Thus, the communicated Web services that are independently developed by different providers, are likely to cause the exchanged data to have a range of different assumptions about the interpretation of their meaning (Li *et al.*, 2011). To exemplify this challenge, consider the term Ticket, this term would refer to a flight ticket if the service is provided by an airline company and would refer to a train ticket if the service is provided by a train company and so on.

Semantic representation: Semantic representation is another serious challenge that stands as a barrier to detecting semantic conflicts between heterogeneous messages. Semantic representation can be thought of as the mapping process between the term, the representation of the term and its meaning. Thus, multiple representations of the same term will lead to misunderstanding in the meaning of the term. It is worthy to note that although, messages can be

exchanged with even multiple representations between the exchanged data, the data itself will by then almost certainly be meaningless or incorrect.

In practice, there are several representations of the same term, for example, the currency can be represented differently from some countries to another, such as Yemen Riyal (YR), Ringgit Malaysia (RM) and so on. This diversity of representations of the same term makes conflicts detection increasingly difficult. There are two factors that for the most part cause this diversity. The first factor is the way in which Web services are independently provided by different providers, who perhaps use different representations of the data (Li *et al.*, 2011). This is due to the way that the implementation of Web services may be adhering to the standards and to the policies of the providers involved. Furthermore, the standards and the policies are likely to differ from one provider to another. This increases the possibility that different representations will have been chosen for the same term. The second factor is, most of Web services are developed for different domains, in which different domains have different data representations (Barnickel, 2011). That means that the challenge of detecting semantic conflicts will increase when the messages exchanged between Web services from different domains.

Message Structure: Web service messages between the interoperating services consist of two main parts the message structure (schema) and the actual message (data) to be exchanged. Message structure often differs from one message to another due to the variety of ways the message may be structured by different developers involved, each with their own different objectives and different knowledge background (Zheng *et al.*, 2006). This all emphasises how there are diverse ways and many possible choices involved in the implementation of Web services (Aragão and Fernandes, 2003). This will all result in there being, at any time, a range of different message structures.

Adherence to a single format for the structuring the messages is neither feasible nor possible. This is because it is not possible for service providers to predict how and where their services capabilities will be consumed. In this way, the difficulty of detecting semantic conflicts is only likely to increase.

3.2. Solution

In practice, a Web service requires to use another services to perform its task (Sahai and Machiraju, 2002) and this done through using the input message that send from Web service sender as an input for the input message

of Web service receiver. WSDL defines the Web service messages as abstract definitions that describe the data being exchanged; the abstraction definitions messages are described in terms of XML-based (Briukhov and Kalinichenko, 2003). Messages consist of parts, which are describing the data of messages in a logical abstract form; these parts transport the data between the message sender and the receiver (Rodriguez *et al.*, 2010).

However, in order to propose a new semantic conflicts detection approach, the aforementioned challenges should be considered and then solved. For this purpose, we involved two components to the proposed approach to mitigate those challenges, which are ontology and semantic conflicts classification. The ontology is responsible for providing semantic interpretations, representations and message structures. Whereas, the classification was chosen to help the tool during detecting the conflicts, which will be based on the semantic information that should be provided by the ontology.

Ontology: Ontology is thought of as a tool that helps to clarify the semantics of information (Li and Ling, 2004), in which the concepts of this information and the relationships between these concepts are formally representing (Alamgir and Mohayidin, 2009). Ontology has the most important impact in the information exchange process (Terzi and Vakali, 2003), this is because, ontology provides the fundamental technology, which is necessary for supporting semantic interoperability (Cheng *et al.*, 2009). In fact, we chose the ontology to be as the backbone of our detection approach; to improve the achievement of semantic interoperability between Web services messages. This is due to the fact that, ontology has shown its ability in interweaving human and machine understanding (Della *et al.*, 2005; Fensel, 2004). Furthermore, the necessary information about semantic interpretation of terms, representation of terms and message structures can be derived from the ontology.

Several languages have emerged to create the ontologies. Despite the success of these languages, Web Ontology Language (OWL) was chosen as a language for implementing the used ontology in our approach. The reason is that, OWL has become a standard language for developing ontology (Grau *et al.*, 2008). OWL is an ontology language that introduced by OWL working group as a World Wide Web Consortium recommendation (W3C, 2004). It represents the meaning of classes (concepts) and the relationships between those classes (Mousavi *et al.*, 2010). In our approach the relationships are very important in order to determine the semantic of the messages elements.

3.3. Semantic Conflicts Classification

As we mentioned previously, our approach also intend to identify the type of the detected semantic conflicts. Accordingly, the identification process should be done based on a clear guideline in order to identify the detected conflicts precisely. In our point of view, the guideline should be able to provide a clear distinction between the causes of every conflict. For example, a customer from Yemen wants to buy a laptop from Malaysia and the output message that sent from Web service sender uses the Yemen Riyal (YR), while the input message of Web services receiver uses the Malaysian Ringgit (MR). As we can see, the difference between those two messages is the currency units; therefore, the guideline should give the relevant type of conflict, which is in this example unit conflict type.

From our approach perspective, we use semantic conflicts classification as a guideline for identifying the type of semantic conflict. The reason is that, the classification associates every conflict type with its relevant cause. Nevertheless, plenty of classifications have been proposed with the aim of classifying semantic conflicts such as (Aragão and Fernandes, 2003; Nagarajan and Verma, 2007). However, Message Level Heterogeneities classification that introduced in (Nagarajan and Verma, 2007) was adapted in our approach. The reason for that, this classification is the most suitable one that suite the need of our approach. Moreover, this classification focuses on the semantic conflicts that may arise at the message level of Web service.

4. RELATED WORK

Semantic conflict detection has attracted researchers from different field, where semantic conflicts may exist. Such as, aspect-oriented programming (Bergmans, 2003), database (Sattler *et al.*, 2003; Sudha and Jinsoo, 2004), model-based development (Reiter and Altmanninger, 2007), Web services (Nagarajan *et al.*, 2006; Nagarajan and Verma, 2007; Ying *et al.*, 2008; Li *et al.*, 2009) and so on.

The detection concept is not new in Web services domain since it has been used to detect several problems, which are related to Web services. Some of the problems that are related to Web service that are required detection are, Web service protocols heterogeneity (Ghosh and Dasgupta, 2010), Web services feature interactions problem (Zhang and Yang, 2006; Xu *et al.*, 2011), semantic conflicts between heterogeneous ontologies (Li and Ling, 2004) and semantic conflicts between heterogeneous messages in data level (Li *et al.*, 2009).

Ghosh and Dasgupta (2010) propose formal method to detect the semantic conflicts between protocols during exchanging the data between Web services, where these Web services are described using different ontologies. The process of Web service composition involves interaction of variety messages. These message interactions sometimes require multiple incompatible features, which lead to feature interaction problem. However, some studies proposed approaches to detect this problem such as (Zhang and Yang, 2006; 2007).

Ontologies have been adapted from database field to Web service field in order to support the interoperability between Web services. From technical point of view, these ontologies are different from each other in terms of their model structures and concepts. Thus, semantic conflicts are still exists in ontology-based semantic interoperability due to semantic unification of concepts (Vernadat, 2010). With respect to ontology conflicts, (Li and Ling, 2004) proposes a comprehensive algorithm to detect and then resolve semantic conflicts based on OWL language.

Achieving successful Web service tasks such as composition, discovery and invocation is always hampered by semantic conflicts, which are the result from data heterogeneity between Web services. These conflicts manifest themselves either at schema-level or data-level. Nevertheless, (Li *et al.*, 2009) proposes a technique to detect and reconcile semantic heterogeneity from context perspective during Web service composition. This approach involves the use of COIN ontology, which is a lightweight ontology. The key idea for detecting semantic conflicts was based on extracting all modifiers of ontology concept, then comparing the modifiers values, if they are not equal that means context conflicts is thus determined. However, the correct interoperability in this approach is based on the modifiers availability of the ontology concepts (Mrissa *et al.*, 2007).

To the best of our knowledge, none of the current approaches has efficiently detected semantic conflicts in data-level and schema-level as well. Therefore, any solution for detecting and solving semantic conflicts should consider both levels in terms of detection and solution. We argue that achieving successful semantic interoperability for Web services messages is based on detecting and then solving semantic conflicts in both level schema and data level.

5. THE PROPOSED APPROACH

The conceptual design for our approach is outlined in **Fig. 2**.

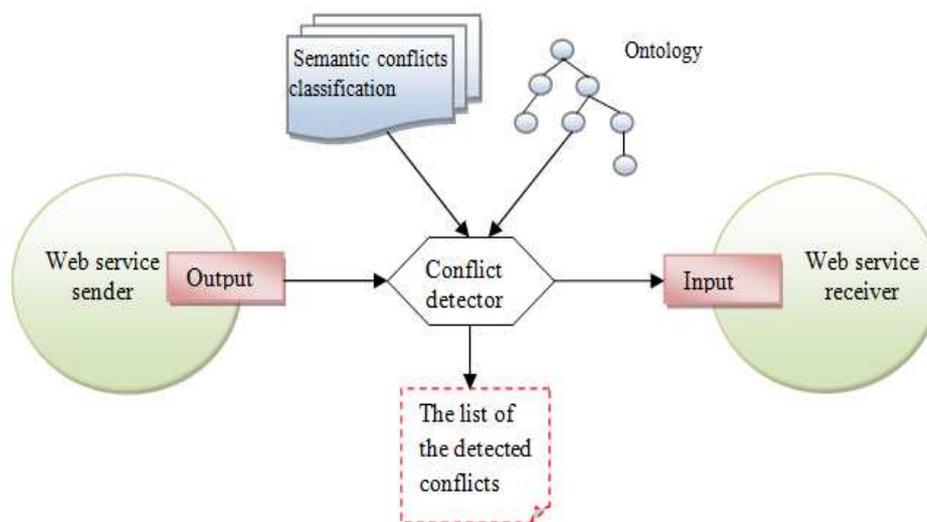


Fig. 2. The conceptual design of the ontology-based semantic conflict detection approach

From our approach perspective, in order to detect the conflicts between two heterogeneous messages, a communication between their Web services should take place. This is due to the fact that the communication between Web services is based on message exchange mechanism (Kona *et al.*, 2009). As can be seen from **Fig. 2**, the main components of our approach are Web services, conflict detector, an ontology and semantic conflicts classification. Two types of Web services are participating in our approach, these being the Web service sender and the Web service receiver. The communication between Web services is based on message exchange mechanism (Chen and Chang, 2007). In order to establish a communication between these Web services, the Web service sender starts the communication by sending a message (the output message) to the Web service receiver. The data from the output message that was sent by Web service sender will be then used as an input for the input message of Web service receiver. The ontology is used with the aim of providing the semantic information about Web services messages to the conflict detector, hence, it is considered as the backbone of our approach. While the conflict detector will be guided by the semantic conflicts classification, since it will support the detector during semantic conflict type identification.

Conflict detector will be responsible for detecting all possible conflicts between Web service messages when messages exchange takes its place. As a consequence, the conflict detector is laid between the output message and the input message of the communicated Web services. This approach assumes that all messages are exchanged using SOAP protocol, this is due to the fact that, SOAP is

the famous protocol at message level (Nezhad *et al.*, 2006). However, detecting semantic conflicts involves around two phases design phase and runtime phase.

Design phase: In this phase we identify the communicated messages in order to detect the conflicts that may arise during messages exchange. Moreover, we create the mapping knowledge between messages elements and the ontology concepts (for more details about mapping knowledge we refer readers to (Al-Baltah and Ghani, 2013). Mapping knowledge provides the semantic of messages elements through the ontology. In addition, in our point of view, the mapping knowledge provides the semantic detector with the differences between messages elements. It is worthy to point out that, the used ontology should capture the domain of the communicated messages in order to semantically describe the messages elements.

Runtime phase: In this phase, semantic conflicts will be detected and the type of conflicts will be identified.

The overview of the conflict detector is depicted in **Fig. 3**. The process of semantic conflicts detection consists with seven steps as below:

Step 1: Select the communicated messages. In practice, most of the real Web services have more than one message, where performing specific task requires specific messages to be exchanged. Therefore, it is very important to determine the specific messages, which are necessary to accomplish the desired task from establishing this communication. From the aforementioned scenario, booth USGeoCoding5 and GeoPhone consist of five messages; however, only one message from each Web service was selected to achieve the scenario task (sending mails to customers).

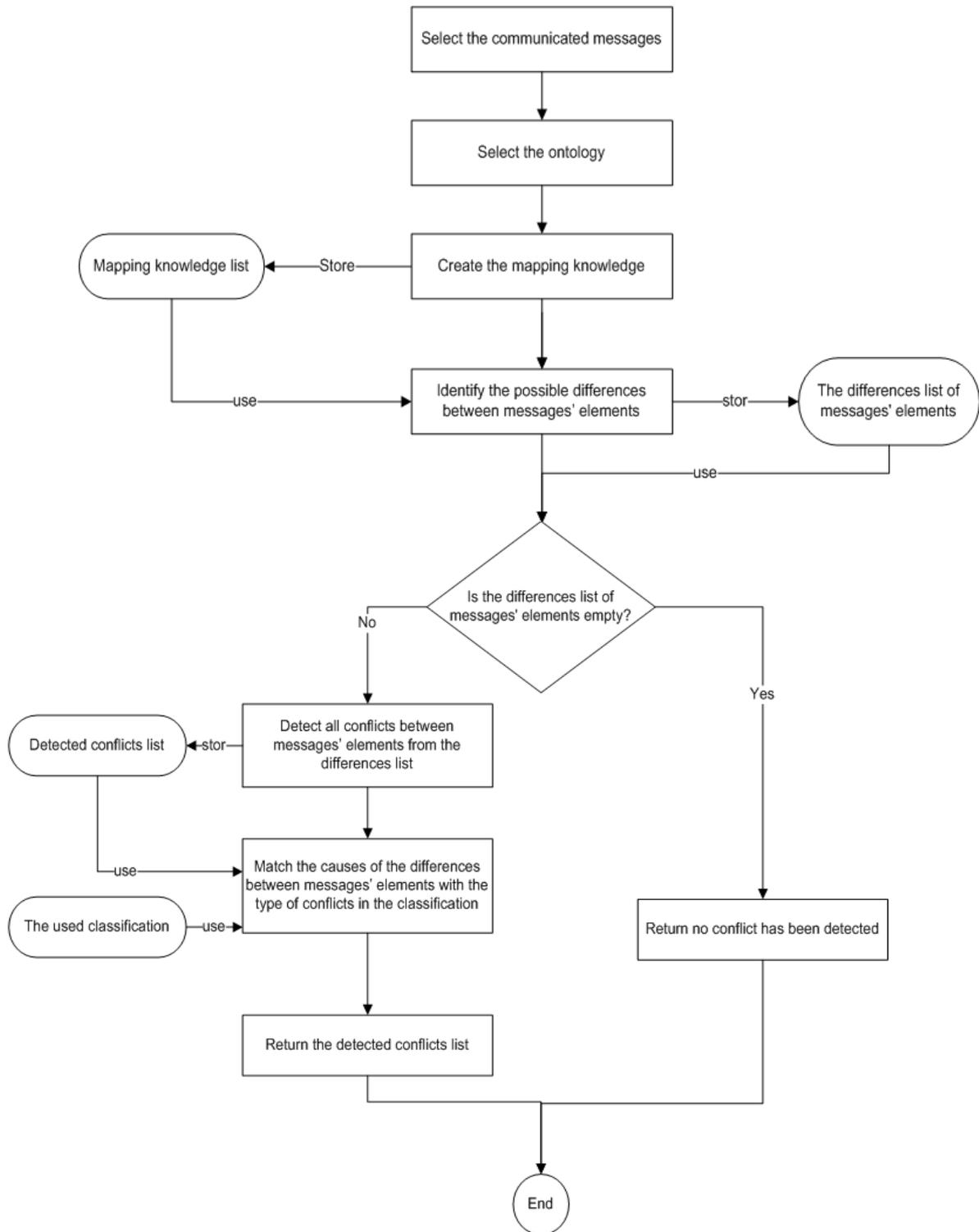


Fig. 3. The process of semantic conflicts detection

Table 1. Summary of the detected semantic conflict types from the scenario

USGeoCoding5s' elements	GeoPhones' elements	Differences	Conflict type
CityStateZIPCode	City, State and Zip	(a) The names are different (b) CityStateZIPCode is more general than City, State and Zip	(a) Naming conflict (b) Generalization conflict
Addressline1, Addressline2	Address	(a) The names are different (b) Address is more general that Addressline1, Addressline2	(a) Naming conflict (b) Generalization conflict

The selected messages are then GetGeocodeUSA from GetGeocodeUSA and GetContactInfo from GeoPhone.

Step 2: Select an ontology; this ontology will be used as a tool to provide the meaning of the messages elements. The participated ontology in the detection process should be the domain ontology of the interoperating Web services. This is because different domains have different ontologies; therefore, the involved ontology in the detection process should describe the semantic of the interoperating Web services. As for USGeoCoding5 and GeoPhone Web services, LSDIS_Finance ontology (http://lsdis.cs.uga.edu/projects/meteor-s/wsdl-s/ontologies/LSDIS_Finance.owl) was used, since both Web services are semantically described using this ontology.

Step 3: Create the mapping knowledge between ontology concepts and the elements of the input and output messages. These mappings are very important to indicate the relationships between elements of the output and the input messages through the ontology. To create the mapping knowledge, the domain expert should specify the corresponding ontology concepts for the output and the input elements of the participated messages. This mapping knowledge will be created only once and it will be used whenever semantic conflicts detection is required. For example, CityStateZIPCode (from USGeoCoding5) is the corresponding element for the elements City, State and Zip (from GeoPhone). Thus the domain expert should map these elements to the relevant ontology concept in order to allow our approach to detect and identify the type of the conflicts between the mapped elements.

Step 4: Identify the possible differences between the elements of the output message and the corresponding elements of the input message based on the ontology. Determining whether the elements of the messages are semantically related or not, will be based on the relationships between the corresponding concepts from the ontology. Furthermore, the relationships between ontology concepts will indicate to what extend these elements are semantically related.

Step 5: Detect semantic conflicts between the elements of the communicated messages from the

differences list. In this step, if there is any difference between the messages' elements will be considered as a conflict. As can be seen from the abovementioned scenario, there are differences between City, State and Zip elements and CityStateZIPCode element, since they are representing using different names and different structure. Also, there are differences between Addressline1, AddressLine2 and Address, because they are representing using different names and different structure. Thus, these differences will be considered as conflicts.

Step 6: Match the causes of the differences between the elements of the communicated messages with the type of conflicts in the classification. This is due to the fact that, every semantic conflict type has a cause that arise this particular conflict type. Therefore, the detection process aims at detecting and identifying the type of the conflict that being detected. As a consequence, the aim of this step is to identify the type of conflicts that being detected in the previous step. Therefore, for any given difference between messages elements from the difference list, there is a relevant conflict type in the classification, which associates with the given difference. To identify the conflict types that exist in the mentioned scenario, the detected conflicts list from step 5 and the used classification in our approach were used. For every detected conflict type we match every cause of the differences with the type of conflicts in the classification. The results of this step were as summarized in **Table 1**.

Step 7: Return the detection result. The result of semantic conflict detector is either no conflict found, or detected conflicts list, which consists the conflicts that have been detected with their conflicts type. With respect to the scenario, the last column in **Table 1** is the detected conflicts list that our approach should returns.

6. DISCUSSION

To cope with semantic conflict problem in the messages of Web services, the solution must firstly detect the conflict and identify the type of the conflicts that being detected in a sound manner. Therefore, the proposed approach copes with the problem from

different two perspectives. Firstly, our approach makes use of both ontology and conflicts classification, in which the ontology is used as semantics provider for the messages element and the classification is used as an aid for identifying the type of conflict. This point gives the strength and the flexibility of adapting our approach in any domain, since our approach is domain independent. For example, to adapt our approach in healthcare domain, the necessary requirement is adapting an ontology that semantically describes the communicated messages from the healthcare domain. In other word, our approach can be used in different domains by adapting the relevant ontology that captures the semantic of the messages from this particular domain.

Secondly, our approach is able to detect the conflicts that would arise due to the use of different interpretations and representations of the same term and due to the use of different message structures. Besides, it identifies the type of the detected conflicts, which add another advantage to our approach. In fact, the process of solving semantic conflicts is commonly called as data mediation, in which every conflict type may require special transformation process. Thus, the result of our approach will provide the necessary information (semantic conflicts and their types) to the mediation; therefore, data mediation will choose only the specific transformation process for the specific conflict type.

Nevertheless, the effectiveness of the proposed approach relies on the selected ontology and the adapted classification. On one hand, the selected ontology in our approach should be a common ontology because its contents are heavily influenced by the semantic conflict types (Kahng and McLeod, 1998), hence, it will provide the proposed approach with the necessary information that is required during semantic conflicts detection process. On the other hand, to the best of our knowledge, the adapted classification is the most suitable one, since it provides a clear conflict reason for every conflict type and covers most of the semantic conflict types that may arise between heterogeneous Web services at message level. As a result, the proposed approach will be able to detect and identify the semantic conflicts effectively, unlike others approach (e.g.,) (Li *et al.*, 2009) which consider only some conflict types.

7. CONCLUSION

Research on semantic conflicts detection is still needs further investigation and improvement to grow to its maturity. This study investigated the important and the need of new conflicts detection and hence, introduced an

ontology-based approach to detect semantic conflicts of heterogeneous Web services at message level.

The proposed approach lies between the input message and the output message. Thus, when messages exchanged is required, the conflicts detector takes both input and output messages as input; and the result is either no conflicts has been detected, or a list that consists all conflicts that have been detected. Furthermore, the detected conflicts will go through under another process in order to identify the conflict type for every detected conflict.

The main components of our approach are ontology and semantic conflicts classification. The ontology is responsible for providing the semantics of the messages elements, which is used during conflicts detection. Whilst the classification is used during the conflict types identification, which is responsible for providing the conflict type for any given difference that cause the conflict.

There are some significant issues to be carried out in the future. Firstly, evaluate the semantic conflicts classification from two perspectives completeness and accuracy. This is will improve the completeness and the accuracy of the detection approach. Secondly, apply the proposed approach in different domains in order to ensure its applicability in different domains.

8. REFERENCES

- Al-Baltah, I.A. and A.A.A. Ghani, 2012. Towards a systematic approach for solving semantic conflicts in heterogeneous Web service messages. Proceedings of 2nd International Joint Conference on Advances in Signal Processing and Information Technology, (PIT' 12), Dubai, UAE.
- Al-Baltah, I.A. and A.A.A. Ghani, 2013. Towards ontology driven semantic conflicts detection in web services at message level. *Int. J. Web Portals*, 5: 71-80. DOI: 10.4018/ijwp.2013070105
- Alamgir, M. and M.G. Mohayidin, 2009. Data mediation to message level conflict in heterogeneous web services. *Int. J. Comput. Internet Manage.*, 17: 41-47.
- Albreshne, A., P. Fuhrer and J. Pasquier-Dorthe, 2009. *Web Services Technologies: State of the Art*. Université de Fribourg, Fribourg, pp: 43.
- Aragão, V. and A.A. Fernandes, 2003. Conflict Resolution in Web Service Federations. In: *Web Services-ICWS-Europe*, Jeckle, M. and L.J. Zhang, (Eds.), Springer, Berlin Heidelberg, ISBN-10: 3540201254, pp: 109-122.

- Barnickel, N. and M. Fluegge, 2010. Towards a conceptual framework for semantic interoperability in service oriented architectures. Proceedings of the 1st International Conference on Intelligent Semantic Web-Services and Applications, Jun. 14-16, New York, pp: 1-7. DOI: 10.1145/1874590.1874602
- Barnickel, N., 2011. Interoperability in eGovernment Through Cross-Ontology Semantic Web Service Composition. 1st Edn., GRIN Verlag, ISBN-10: 3640942744, pp: 93.
- Bergmans, L.M.J., 2003. Towards detection of semantic conflicts between crosscutting concerns. University of Twente.
- Blake, M.B. and H. Gomaa, 2005. Agent-oriented compositional approaches to services-based cross-organizational workflow. Decision Support Syst., 40: 31-50. DOI: 10.1016/j.dss.2004.04.003
- Briukhov, D. and L. Kalinichenko, 2003. Extension of compositional information systems development for the Web services platform. Proceedins of the 7th East European Conference on Advances in Databases and Information Systems, Sept. 3-6, Springer Berlin Heidelberg, Dresden, Germany, pp: 16-29. DOI: 10.1007/978-3-540-39403-7_4
- Chen, W. and E. Chang, 2007. An analysis of web services mediation architecture and pattern in synapse. Proceedings of the International Conference on Advanced Information Networking and Applications Workshops, May 21-23, IEEE Xplore Press Niagara Falls, pp: 1001-1006. DOI: 10.1109/AINAW.2007.76
- Cheng, Z., H. Keqing and B. Li, 2009. A framework to support interoperability among Web service registries. Proceedings of the International Conference on Computer and Information Science, Jun. 1-3, IEEE Xplore Press, Shanghai, pp: 891-896. DOI: 10.1109/ICIS.2009.62
- Christensen, E., F. Curbera and L. Bing, 2001. Web Services Description Language (WSDL)
- W3C, 2004. OWL web ontology language overview.
- Della, V.E., D. Cerizza, V. Bicer, Y. Kabak and G.B. La *et al.*, 2005. The need for semantic web service in the eHealth. Proceedings of the W3C Workshop on Frameworks for Semantics in Web Services, Jun. 9-10, Innsbruck, Austria.
- Domingue, J., L. Cabral, S. Galizia, V. Tanasescu and A. Gugliotta *et al.*, 2008. IRS-III: A broker-based approach to semantic Web services. Web Services Agents World Wide Web, 6: 109-132. DOI: 10.1016/j.websem.2008.01.001
- Fensel, D., 2004. Ontologies: A Silver Bullet for Knowledge Management and Electronic Commerce. 1st Edn., Springer, Berlin, ISBN-10: 3540003029, pp: 162.
- Ghosh, P. and P. Dasgupta, 2010. A formal method for detecting semantic conflicts in protocols between services with different ontologies. Proceedins of the International Conferences on Recent Trends in Networks and Communications, Jul. 23-25, Springer Berlin Heidelberg, Chennai, India, pp: 553-562. DOI: 10.1007/978-3-642-14493-6_57
- Grau, B.C., I. Horrocks, B. Motik, B. Parsia and P. Patel-Schneider *et al.*, 2008. OWL 2: The next step for OWL. Sci. Services Agents World Wide Web, 6: 309-322. DOI: 10.1016/j.websem.2008.05.001
- Hirst, G., 1992. Semantic Interpretation and the Resolution of Ambiguity. 1st Edn., Cambridge University Press, Cambridge, ISBN-10: 052142898X, pp: 263.
- Kahng, J. and D. McLeod, 1998. Dynamic Classification Ontologies: Mediation of Information Sharing in Cooperative Federated Database Systems. In: Cooperative Information Systems: Trends and Directions. Papazoglou, M.P. and G. Sohlageter (Eds.), Academic Press, San Diego, Calif, ISBN-10: 0125449100.
- Kona, S., A. Bansal, L. Simon and G. Gupta, 2009. USDL: A service-semantics description language for automatic service discovery and composition. Int. J. Web Services Res., 6: 20-48. DOI: 10.4018/jwsr.2009010102
- Li, C. and T.W. Ling, 2004. OWL-based semantic conflicts detection and resolution for data interoperability. Proceedings of the Conceptual Modeling for Advanced Application Domains, Nov. 8-12, Springer Berlin Heidelberg, Shanghai, China, pp: 266-277. DOI: 10.1007/978-3-540-30466-1_25
- Li, X., S. Madnick, Z. Hongwei and F. Yushun, 2009. Reconciling semantic heterogeneity in web services composition. Proceedings of the International Conference on Information Systems, (ICIS' 09).
- Li, X., S. Madnick, H. Zhu, X. Li and S.E. Madnick *et al.*, 2011. Improving the data quality of web services composition. Proceedings of the 7th International Workshop on Quality in Databases, (QDB' 09).

- Mousavi, A., M.D.J. Nordin and Z.A. Othman, 2010. An ontology driven, procedural reasoning system-like agent model, for multi-agent based mobile workforce brokering systems. *J. Comput. Sci.*, 6: 557-565. DOI: 10.3844/jcssp.2010.557.565
- Mrissa, M., C. Ghedira, D. Benslimane, Z. Maamar and F. Rosenberg *et al.*, 2007. A context-based mediation approach to compose semantic Web services. *ACM Trans. Internet Technol.* 8: 4:1-4:23. DOI: 10.1145/1294148.1294152
- Nagarajan, M. and K. Verma, 2007. Ontology driven data mediation in web services. *Int. J. Web Services Res.*, 4: 104-126. DOI: 10.4018/jwsr.2007100105
- Nagarajan, M., K. Verma, A.P. Sheth, J. Miller and 2006. Semantic interoperability of web services-challenges and experiences. *Proceedings of the International Conference on Web Services*, Sept. 18-22, IEEE Xplore Press, Chicago, IL, pp: 373-382. DOI: 10.1109/ICWS.2006.116
- Nezhad, H.R., B. Benatallah, F. Casati and F. Toumani, 2006. Web services interoperability specifications. *Computer*, 39: 24-32. DOI: 10.1109/mc.2006.181
- Rathore, M. and U. Suman, 2011. A quality of service broker based process model for dynamic web service composition. *J. Comput. Sci.*, 7: 1267-1274. DOI: 10.3844/jcssp.2011.1267.1274
- Reiter, T. and K. Altmanninger, 2007. Models in conflict-detection of semantic conflicts in model-based development. *Proceedings of the 3rd International Workshop on Model-Driven Enterprise Information Systems, (EIS' 07)*, Funchal, Madeira, pp: 29-40.
- Rodriguez, J.M., M. Crasso and A. Zunino, 2010. Improving web service descriptions for effective service discovery. *Sci. Comput. Programm.*, 75: 1001-1021. DOI: 10.1016/j.scico.2010.01.002
- Sahai, A. and V. Machiraju, 2002. Automated SLA monitoring for web services. *Proceedings of the 13th IFIP/IEEE International Workshop on Distributed Systems: Operations and Management: Management Technologies for E-Commerce and E-Business Applications*, Oct. 21-23, Springer Berlin Heidelberg, Montreal, Canada, pp: 28-41. DOI: 10.1007/3-540-36110-3_6
- Sahin, K. and M. Gumusay, 2008. Service Oriented Architecture (SOA) based web services for geographic information systems. *Proceedings of the International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, Jul. 3-11, Beijing, China, pp: 625-630.
- Sattler, K. U., S. Conrad and G. Saake, 2003. Interactive example-driven integration and reconciliation for accessing database federations. *Inform. Syst.*, 28: 393-414. DOI: 10.1016/S0306-4379(02)00023-6
- Siew, P.L., L.P. Chan and E.W. Lee, 2006. Web services implementation methodology for SOA application. *Proceedings of the IEEE International Conference on Industrial Informatics*, Aug. 16-18, Singapore. pp: 335-340. DOI: 10.1109/INDIN.2006.275822
- Siluvai, I.S.P. and V.J.S. Kumar, 2013. A framework for simple object access protocol messages to detect expansion attacks for secure webservice. *J. Comput. Sci.*, 9: 308-313. DOI: 10.3844/jcssp.2013.308.313
- Sudha, R. and P. Jinsoo, 2004. Semantic conflict Resolution Ontology (SCROL): An ontology for detecting and resolving data and schema-level semantic conflicts. *IEEE Trans. Knowl. Data Eng.*, 16: 189-202. DOI: 10.1109/TKDE.2004.1269597
- Terzi, E. and A. Vakali, 2003. Knowledge Representation, Ontologies and the Semantic Web. In: *Web Technologies and Applications*, Zhou, X., M. Orłowska and Y. Zhang (Eds.), Springer, Berlin Heidelberg.
- Vernadat, F.B., 2010. Technical, semantic and organizational issues of enterprise interoperability and networking. *Annual Rev. Control.*, 34: 139-144. DOI: 10.1016/j.arcontrol.2010.02.009
- Xu, C. and S.C. Cheung, 2004. Semantic Interpretation and Matching of Web Services. In: *Conceptual Modeling-ER*, Atzeni, P., W. Chu, H. Lu, S. Zhou and T.W. Ling, Springer, Berlin Heidelberg.
- Xu, J., K. Chen, Y. Duan and S. Reiff-Marganiec, 2011. Modeling business process of web services with an extended STRIPS operations to detection feature interaction problems runtime. *Proceedings of IEEE International Conference on Web Services*, Jul. 4-9, IEEE Xplore Press, Washington, pp: 516-523. DOI: 10.1109/ICWS.2011.73
- Ying, W., Z. Rubo and J. Wang, 2008. Ontology-Based Heterogeneous Messages Matching in Web Services. *Proceedings of the 4th International Conference on Wireless Communications, Networking and Mobile Computing*, Oct. 12-14, China, Dalian. pp: 1-4. DOI: 10.1109/WiCom.2008.1234

- Zhang, J. and F. Yang, 2007. Immune-Inspired Online Method for Service Interactions Detection. In: SOFSEM 2007: Theory and Practice of Computer Science. J. Van Leeuwen, G. Italiano, W. Van Der Hoek *et al.* (Eds.), Springer Berlin/Heidelberg, ISBN-10: 978-3-540-69506-6, pp: 808-818.
- Zhang, J. and F. Yang, 2006. Detecting the Web Services Feature Interactions. In: Web Information Systems-WISE, Aberer, K., Z. Peng, E. Rundensteiner, Y. Zhang and X. Li, Springer Berlin Heidelberg.
- Zheng, L., L. Shiyan, A. Ghose and P. Hyland, 2006. extending semantic web service description by service assumption. Proceedings of the International Conference on Web Intelligence, Dec. 18-22, Hong Kong, China, pp: 637-643. DOI: 10.1109/WI.2006.74