

Research Article

An Overview of Semantic Web Service Description based on Model Driven Approaches

Alaeddin Kalantari, Suhaimi Ibrahim and Mohammad Ali Nematollahi

Faculty of Computer Science and Information System, Advanced Informatics School (AIS), Universiti Teknologi Malaysia, Malaysia

Abstract: In this study, formal MDA approaches for describing Semantic Web Services (SWS) are discussed. Although syntactic approach is used for fulfill the web service task, the main concentration of this study is on SWS. Basically, MDA approaches for SWS are divided into four categories such as software methodology based, UML formalist based, formal-based and query-based. Each of these SWS approaches uses OWL-S, SAWSDL, or WSMO to construct the SWS description. Therefore, each MDA approach is fully discussed to provide comprehensive overview on SWS technology.

Keywords: Model driven approach, semantic web services, UML profile, WSMO

INTRODUCTION

Semantic web is a solution to address the limitation of current web, which enables machine to read and understand by processing the semantics or meaning of resources. Thus, the automation and accuracy of different aspects like information integration, search and information extraction are increased. As described by McIlraith *et al.* (2001), Semantic Web Services (SWS) are defined by bringing semantic into Web services. Indeed, Semantic Web Service describes the capability and contents of a service in machine-interpretable language to facilitate and boost the quality of service selection, discovery, composition, invocation and execution (McIlraith and Martin, 2003). Although Model Driven Architecture (MDA) uses several different models such as CIM, PIM and PSM to facilitate the development of software, writing semantic web service specification through MDA submissions is not easy enough to developer and need additional knowledge in the field of knowledge engineering and knowledge representation. These areas of knowledge and technologies are still unknown for software developers (Timm, 2008). Also, several tools and editors such as OWL-S Editor, WSMO studio (WSMO Studio, 2009) and WSMOViz (Kerrigan, 2006) have been proposed to facilitate writing semantic specifications. However, developers still need to know the concepts and syntaxes of the semantic web service languages. This lack of knowledge and also the difficulty of these languages cause the adoption of

semantic web services is being slow down (Timm, 2008). In order to alleviate this problem, researchers have proposed Model Driven Architecture (MDA) (Miller and Mukerji, 2003) based approaches to automatically extract semantic description of web services from the defined models. Most of them have used UML (Group, 2010) as modeling language in MDA due to its widespread adaption among the software developers (Bendraou *et al.*, 2009). Indeed, Most of the semantic web service development researches such as Grønmo and Oldevik (2005), Timm (2008), Bensaber and Malki (2008) and Kim and Lee (2009) focus on creating models using UML diagrams to model atomic and composite web services semantically. These approaches have provided automated semantic web service description generator. They have separated the semantic description elaboration from the actual development of the underlying services (Brambilla *et al.*, 2007). They have used UML diagrams to provide semantic description of the current web applications instead of creating semantic annotations during the development of the services. Furthermore, most of these approaches have focused to generate OWL-S as outgoing semantic web service specification and in most of them the models are depend on OWL-S specifications. In addition to above efforts, some approaches like Torres *et al.* (2006) and Brambilla *et al.* (2006) and MIDAS-S (Acuña and Marcos, 2006) have proposed the creation of semantic annotations in the way of software engineering methodology. However, they have extracted semantic

Corresponding Author: Alaeddin Kalantari, Faculty of Computer Science and Information System, Advanced Informatics School (AIS), Universiti Teknologi Malaysia, Malaysia

This work is licensed under a Creative Commons Attribution 4.0 International License (URL: <http://creativecommons.org/licenses/by/4.0/>).

descriptions during the development of the services. However, their models are depending on the outgoing semantic web service language and due to this lack of reusability; the models cannot be used to generate other semantic web service languages. On top of that, there is no mapping rule to generate WSMO based specification, as one of the most prominent ontologies, in MIDAS-S approach to illustrate its correctness. The only approach that has proposed the mapping roles is Brambilla *et al.* (2007) approach, which is based on BPMN and WebML and it is different from UML modeling.

The rest of paper is organized as following:

SWS APPROACHES

Figure 1 presents the state-of-the-arts approaches for MDA based approaches. In the follows, each MDA approaches have been fully discussed to provide comprehensive overview on SWS technology.

The most focus of this research is on semantic description of Web services however, in some cases the syntactic methods may be used to fulfil the given tasks. In this section, first, the various SWS approaches are introduced and the most relevant of them to this research are described in detail.

Software Methodology-based: Software methodology-based class is applied software engineering principals and methodological solution for

creating semantic annotations during the services development. This approach is capable to combine web service development and the best practices of the semantic specification. In the follows, the most important studies in this class have been described.

Due to difficulty of modelling ontologies and semantics which are required the target domain deeply, web engineering is applied system domain by using class and activity diagrams (Torres *et al.*, 2006). However, generation these semantic content need to obtain knowledge profile. The main contribution of this approach is providing semantic content and functionality for generation of SWS (Torres *et al.*, 2006). For this reason, Object Oriented Web Solution (OOWS), which is based on web engineering method, has been used to develop SWS. Furthermore, OWL and OWL-S constructions are formed by class diagrams, activity diagrams and a state-chart profile based on domain ontology modeling. In this approach, OOWS is enriched in extended way based on a new dimension (which is called semantic application specification) to construct conceptual schema (behavior and system structure) with high precision and without any ambiguously. Therefore, two OWL and OWL-S models have been developed for specifying the system model and describing external agents respectively. For OWL model, six various ways are available including a class identifier, as an enumeration, as a property restriction, as a union, as a complement or as a pair wise distinct set of individuals. However, for OWL-S just three parts

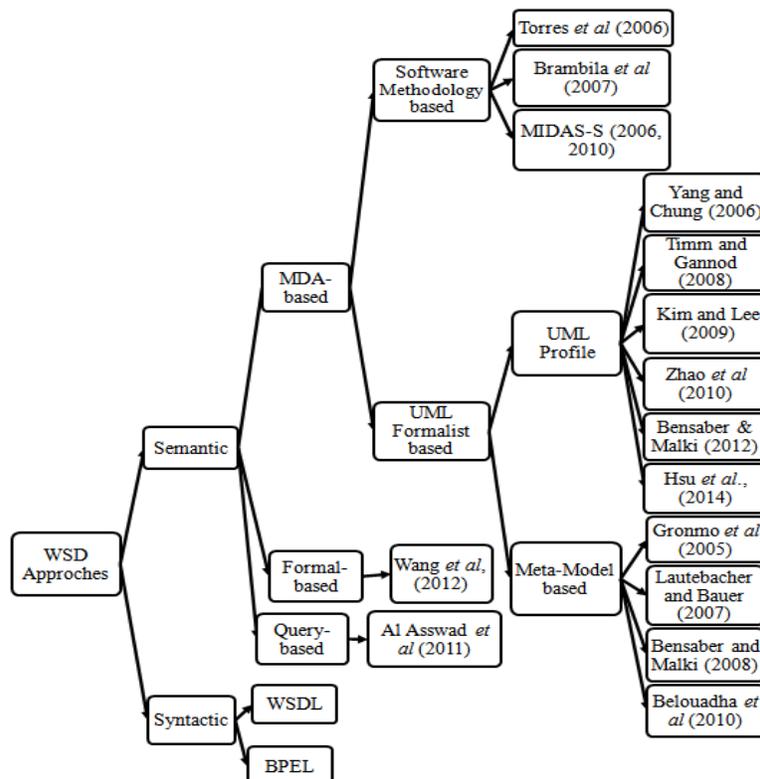


Fig. 1: Classification of related works in MDA approaches

such as service profile, process model and service grounding are available. The methodological solution is proposed in modeling level to improve system functionality in respect to exposing external application usage by extension of web engineering which specify an ontology model. Therefore, two transformation rules based on building and generating of the ontology model has been proposed. Although UML stereotypes could not show UML profile graphically, separation ontology meta-model to general concepts and concrete instances could automatically generated from the knowledge from persistence layer which is based on a database. Finally, it can be concluded that automatic and manual processes should be combined to model SWS. In order to model SWS, three steps should be performed. Firstly, system functionality is selected based on class diagrams. Secondly, transformation is performed on selected in last step. Thirdly, complete service description by specified and necessary elements of source model. Therefore, the main advantage of this approach is using structural and behavioral models such as activity, class and state-chart diagram to provide functionalities and operation for external systems. However, unfortunately the service profile should be created by hand.

MIDAS-S, 2006 and 2010: These SWS approaches are WSMO-based which is developed Web Information System (WIS). The semantic aspects of PIM and PSM are UML-based which can be integrated to other aspects such as hypertext, content and behavior. WSMO logical axioms are represented by UML profiles based on OCL. All the top WSMO elements such as ontologies, mediators, goals and web services are modelled in PSM level. Except mediators, other elements are classified to two main models such as context and content models. The UML diagram is enlarged by some information about namespace, used mediators and imported ontologies.

MIDAS-S, 2006: A SWS approach is proposed through a case study by extension of MIDAS (Known as MIDAS-S). Basically, MIDAS is a model-driven methodology for development of Web Information System (WIS). MIDAS has been proposed to model WIS based on two orthogonal dimensions which are typical aspects of web application modeling (i.e., content, behavior and content) and the degree of platform dependency. However, MIDAS-S has been proposed based on associated WSML and UML to capable development of the SWS with only knowledge of UML modeling.

In this approach the main concentration is on the PSM level of semantic aspect of MIDAS-S. For this reason, Ontology Definition Meta-Model (ODM) has used in semantic aspect of MIDAS-S for modeling semantic web ontology language based on MDA. ODM can improve platform independent ontology models and also can model the mapping rules to various specific

ontology models. This approach can take the UML graphical modeling capability by defining different UML profile and editing WSMO elements. However, generating WSML description of UML model required mapping rules which is not addressed in this study. In this approach, WSMO model is split into ontology context and content models. Ontology context model can perform extension on UML package diagram and it also collects ontology information such as used mediators, namespaces and imported ontologies. However, ontology content model can represent various modeled ontology elements such as attributes, axioms, concepts and instances. Similar to the WSMO ontology modeling, SWS modeling also applies two models including web service context and content models. Therefore, UML package diagram is extended by web services context model and collected information of the web services context.

Compliant MIDAS-S 2010: In order to develop SWS fully UML-compliant, Object Constraint Language (OCL) (De Castro *et al.*, 2006) is applied to represent WSMO axioms. OCL can provide high UML model expressivity which specifies invariant conditions over objects of the model. Furthermore, OCL is very easy for users who don't have strong logical or mathematical background to read/write due to belong to UML family. Therefore, both WSML and OCL are similar to define notations based on first-order logic. In order to define axioms by OCL based on WSMO, three case studies have been investigated including specifies a new concept, specify the capability and specify a relation of a SWS. It seems OCL can provide language closer than WSML for developers. Furthermore, OCL can represent logical expressions without knowing the rules of WSML.

PISA MIDAS-S 2010: This approaches has been applied PISA which provides software architecture techniques follows MDA approach at PIM level that take to account data and behavior of web portals. In this approach, services have been split into two groups: access and core services groups which are used to wrap the external services and achieved integration mandatory services respectively. The user's requirements are translated to goals as WSML messages which then is executed by WSMX. As WSMX is discovered the web services which have the best matches with that goal. Next, data service requirement is mediated and transformation between source format ontology and discovered web services is done by mapping rules. Lastly, invokes and formats this transformation results. Basically, PISA MIDAS-S has applied SOD-M (Service Oriented Development Method) (De Castro *et al.*, 2006) for behavior development and HM3 (Hypertext Modeling Method or MIDAS) (Cáceres *et al.*, 2005) for Hypertext development.

To overcome the difficulty of semantic description in current web semantic approaches, semi-automated generator is proposed to extract semantic descriptions from the application design in WSMX environment (Brambilla *et al.*, 2007). In order to develop semi-automated WSMO-based SWS, this approach has applied Business Process Model and Notation (BPMN) and web modeling language (Brambilla *et al.*, 2007). Both BPN and web modeling diagrams have been serialized based on an XML serialization by XSLT as a transformation language. From this XSLT model, the description of SWS is generated based on WSMO. Before WSMX execution, CASE tool WebRatio (Margaria *et al.*, 2007) is used on web modeling language to convert into an executable code.

In order to transform high-level models into software components, reengineering of software components has been applied to develop SWS requirements which offer rich model description. Although some of studies have separated SWS description from actual development, the creation of SWS annotations should be occurred with SWS development at the same time. Furthermore, three techniques and notations have been developed to cover various aspects of design including high-level design of the global choreography, design of the underlying data model of the cross enterprise applications and design of web service interfaces of an integration platform and applications. Basically, four phases have been suggested by means of development process including process design, data design, hypertext design and semantic description. However, the main concentration of this approach is on semantic description for generating semi-automatically WSMO-compliant semantic SWS specification. Therefore, a new design phase has been proposed for WSMO compatibility. In business process design, each task is mapped to services based on internal, external, or semantic invocations and they are represented by BPMN (or equivalent). In data model design, ontologies are extracted based on four steps. Firstly, existing remote ontologies are imported from remote sources manually. Secondly, WebML data model is translated into WSMO-compliant ontology. Thirdly, process ontology is extracted from BPMN specification. Fourthly, the existing ontologies concepts in the first step can be annotated by WebML data and BPMN model. In querying ontologies design, enhanced WebML has been exploited to navigate ontology instance by modification of primitives and introducing the new ontologies. In service and user interfaces design, both WebML and BPMN diagrams, which have been stored as XML serialization, are transformed into a XML WebML model by serialized XSLT.

During all these phases, description of SWS has been driven from the specification of design by WSMO-compliant semi-automatically. Four WSML components are required to extract semantic description

from the application design including WSML goal, WSML web services choreography, WSML web services capability and WSML wwMediator:

WSML web services capability: Based on behavior of inputs and outputs of the services provided by the WebML and BPMN models of web services, pre/post conditions are extracted semi-automatically. Some inexplicit capabilities and assumptions, which are hidden in web services implementation, must add manually.

WSML web services choreography: Services choreography is required some annotation, which is provided by designer, to establish all possible interaction sequences of the service. By analyzing the invocation order of the various service operations in BPMN model, choreography sequences can be extracted automatically.

WSML goal: By combination of information from WebML and BPMN levels, goals are extracted automatically. Therefore, objects, conditions and sequences of web services in BPMN diagram have combined with webML hypertext models in order to analysis semantics which are embedded in composition and navigation of the pages. Then, goal of the web services calling is generated based on semantic description of the web services which is invoked by user or machine.

WSML wwMediator: Possible interactions, such as services choreography, parameter specification and basic interface, which are in high-level BPMN description, can be extracted for designing of the mediation services. As a result, the mediator can be generated automatically by transforming every BPMN activity from predefined operations' chain.

For implementation, both BPMN editor and WebRatio were applied to skeleton the prototype which allows generating WebML hypertext model based on BPMN diagrams automatically. It should be mentioned that generation of XML documents of the WebML models were transformed by XSLT technology. Error! Reference source not found. Shows overall phases for the extraction of SWS.

UML Formalist-based: According to OMG, there are two UML extensions namely: UML profile and Meta-model. In contrast of meta-model that extend the UML through Meta Object Facility (MOF), UML profile is extended by the collection of specialization such as tagged values, constraints, stereotype and notation.

UML Profile: In UML profile approach, SWS description extracts from UML diagrams which are provided by UML tools. Furthermore, this approach separates semantic descriptions from development of actual underlying services.

Due to difficulty and time-consuming process for creating service ontologies, this task should be developed effectively and automatically. Although some ontology editing tools are available (Yang and Chung, 2006), the developers require to have OWL-S knowledge. However in this approach, OWL-S service model ontology is generated based on UML state-chart and class diagrams. UML is common software engineering modeling technique for most of the developers. Since service model ontology provides crucial information about interoperation among services, this study is focused on service model ontology in OWL-S which is classified into two sub-processes: atomic and composite services generation. The atomic service uses UML class diagrams but the composite service uses UML state-chart diagrams to generate information. Furthermore in composite services, mapping UML diagrams and OWL-S specification are based on a set of the transaction rules. Similar to UML diagrams that represents conditions by GUL, OWL-S specifications uses SWRL to represent the conditions. It should be noted that XSLT transformation language is applied to mapping XMI documents to OWL-S specifications. This approach is complemented the previous study (Paolucci *et al.*, 2003) which could not map automatically between WSDL document and OWL-S ontology for composite process. Furthermore in this approach, the conditions can be express with UML (as GUI standard for software modeling) which is easier than OCL. For generating OWL-S service model ontology, UML class diagrams can provide atomic services information and their properties like IOPE (Input, Output, Precondition and Effect). Furthermore, information about composition services can be retrieved from UML state-chart diagram. Basically, state-chart diagram is defined semantic information better than class diagram. Furthermore, it can provide some basic information about flow constructs including conditional branching, sequence, structured loops, synchronization and concurrent threads which are primitives in many process modeling languages. Some mapping rules for UML state-chart diagram primitives such as sequences, Split, Split+Join, Choice, AnyOrder, If-Then-Else, Iterate, Repeat-While and Repeat-Until have been proposed. Although there are some GUI expressing condition languages for OWL-S (such as SWRL, KIF, RDF and PDDL), this approach has selected SWRL due to high standard and top OWL level. For transformation phase, UML class and state-chart diagrams are exported into the two XMI files separately which are fed into two XSLT applications (more detail in McIlraith and Martin (2003) to produce output files. For validation of OWL-S service ontology generation, an available site (McIlraith *et al.*, 2001) in W3C has been applied to test RDF-level. Furthermore, OWL validators (Fensel *et al.*, 2011; Berners-Lee *et al.*, 2001) have been introduced to support beyond RDF level.

This approach is based on UML class and activity diagrams as well as their UML profile to develop SWS in OWL-S (Timm, 2008). The tasks of execution, grounding and specification are fulfilled by SWS and WSDL. For this reason, four main phases including modeling, conversion, grounding and execution have proposed. UML class and activity diagrams can modeled the service' structure and composition respectively. Although some advantages of OCL for symbolizing conditions of UML diagrams have taken, these conditions should be transformed into SWRL based on OWL-S serialization via XSLT. To provide grounding, description and execution for WSDL documents with any number, an automatic Specification and Execution (SPEX) tool has used. Due to develop an approach for SWS description without any knowledge about formal software method, require knowledge of developer shouldn't beyond UML modeling language. Therefore, a bridge has been created to facilitate incorporation both web services and SWS. It means a MDA technique must be integrated and generalized SWS into composite SWS by OWL-S specification. Moreover, the details of SWS should be hid to allow developer focusing on designing SWS compositions. For these reasons, a framework has been proposed to compatible with WSDL which concepts are mapped into groundings OWL-S concepts automatically. The proposed framework has modeled three specified layers (PSM0, PSM1 and PSM2) in PSM. The creation of first layer (PSM0) is adding OCL and semantic extensions to UML activity and class diagrams. The second layer (PSM1) is transformation of PSM0 into OWL-S specification. The third layer (PSM2) is grounded of second layer (PSM1) by using SPEX tool. For modeling, the developer uses UML profile to facilitate creation of model which aids to transform to OWL-S specification documents (with RDF/XML syntax) in conversation process. In addition, OCL conditions are transformed into Semantic Web Rule Language (SWRL). In grounding process, OWL-S specification is mapped into WSDL-based web services by SPEX tool. This mapping required two types of mapping. First, OWL-S atomic processes should be mapped into WSDL an operation which is one to one mapping. Second, each OWL-S input or output concept is mapped into to a WSDL input or output message part respectively which may also convert data type. Finally in executing process, monitored and feedback are performed on services in real-time manner.

A model-driven approach has proposed to extract SWS description from complex UML diagrams without OWL-S grammar knowledge (Kim and Lee, 2009). In other words, OWL-S specifications are generated from UML class diagram (domain ontology representation) and UML activity or sequence diagrams (representation the behavior of business process). Furthermore, different features of OWL-S are supported by UML

profile. Basically, this approach is based on three phases including: ontology modeling, process modeling and transformation modeling. Firstly, imported domain ontology is presented by a class diagrams. Secondly, the process behavior is expressed by UML sequence or activity diagrams through UML profile (such as stereotype, tagged value and constraint). Thirdly, UML diagrams generate XMI documents which are mapped to OWL-S description by using XSLT script. Each of these steps is discussed in the details in the follows:

Ontology modeling: Definition of the input and output in UML operations and activities are required to link with ontology concept. Although development formal OWL-based domain ontology is needed to represent concepts in each application domain, ontologies are imported as class diagram automatically in this approach. However, class diagrams may be built manually when relevant ontologies are not existed. Therefore, a class diagram can represent constituent properties and hierarchical relationships among ontological concepts. The name and attribute of the UML class are defined based on concepts and properties of the ontology.

Process modeling: Due to visualization ability of UML to show behaviors and functions of the business system, processing is applied on sequence and activity diagrams of the UML to transform them into OWL-S representation. In processing a sequence diagram, the main task is identifying the system's behavior through a set of sequential interactions among objects. Operations have been mapped into atomic processes in OWL-S. Although complex behaviors (such as conditional, iteration, critical interaction) don't have any corresponding constructs in OWL-S, this approach can transform some operators (such as alt, loop and par) into OWL-S control constructs. In processing an activity diagrams, system's workflow which control flow from an activity to another activity, is transformed into OWL-S control constructs.

XSLT: After XMI file is extracted from UML diagrams, XMI should be transformed to OWL-S ontology by using XSLT script. However, conditional statement in OWL-S ontology may transform into Knowledge Interchange Format (KIF) or Semantic Web Rule Language (SWRL).

Actually, reusing UML models which have already existed, can construct structure on top level. In other words, the most interest of this approach is concentration on creation of the models without considering writing codes. This approach can solve the problems of the control flow of complex processes by control constructs of OWL-S.

In this approach, a transformation between high-level modeling and low-level description language has

been proposed for SWS (Zhao *et al.*, 2009). In order to achieve understandability, expressiveness and machine-processable for SWS, a frameworks has developed by using three separate parts. In the first part, SWS is modeled by UML which may be extended via UML profile. Due to difficulty in processing the UML diagrams, XMI standard has applied for serialization of the UML model to XML files which are more processable. Actually, UML meta-model is defined to model non-functional properties (neither behavioral nor functional) of SWS through tagged value which adds extra values into the model. Normally, tagged value has a pair which is attached to stereotypes. Stereotypes should be defined for each WSML elements (ontology, goal, mediator and web service) based on their specific keywords. For better understandability, these WSML elements are described by a tree (called syntax tree). However, some of the WSML construct (like functionality and capability) should not be modeled as a class. This functionality/capability is considered as simple information due to lack of complex information in its signature.

In the second part, UML model is transformed into WSMO source code via XSLT which transformation rules are defined according to conceptual mapping. XSLT has five templates for searching for top level node and sub-elements in syntax tree, for handling non-functional properties as well as namespaces and for transforming the activity nodes. Later on, OCL expressions and logical details can be transformed and then added to framework. OCL is used to ensure non-functionality format by restricting the attribute values. In order to avoid manual transformation which may cause error, one to one mapping is preformed to provide consistency between implementation and design. Furthermore, reverse engineering can be used for backward transformation from WSML to UML model. In the third part, the entire procedure is complete. The syntax conversation is performed by WSML, which convert human readable WSML syntax into XML description. Basically, each WSML syntax has two main parts including logical expression syntax and conceptual syntax. Although logical part has five different language variants, it is same for the conceptual part which is necessary for molding services in stability manner. Consequently, UML profile is focused on conceptual syntax. Several elements such as elements, data types, relationships, transition rules and properties need to map. Although several primitive WSMO data types can easily be mapped, defining new types are required for the rest of WSMO data types. The transition rules in WSML choreography can be modeled by UML activity diagram which is divided into four types including update rule (fact deletion, fact update and fact addition), flow control rules (if-then rule), choose rule and for-all rule. These rules are transformed into UML action, conditional, expansion region. Logical expressions can express by UML

constraint. Furthermore, they are applied to put more constrain on web service ontology. Due to WSML defines specific logic syntax for itself, there is no direct match between WSML and OCL which OCL expressions should be defined them.

This approach extended the UML profile of the class diagram for specifying SWS (Bensaber and Malki, 2012). To achieve this aim, two UML meta-models (including target and source) have created. The source meta-model (UML model) is transformed into the target meta-model, (WSMO specification). In other word, ALT transformation has applied to map XMI specification of the UML model into WSMO specification. Actually, UML profile is extended the capability of UML modeling language by a collection of stereotypes, custom data and tagged values. Not only stereotypes can provide distinction among various classes' types but also, it creates a top level standard over UML class for modeling constructs by a Meta modelling language. Furthermore, developer can attach some pairs (name and value) to UML model by using tagged values. Different constructions of the WSMO, which conjugate with UML static structure, are modeled by these UML profile elements in PIM level for facilitating better WSMO transformation. The proposed meta-model UML profile has two parts: UML standard elements (such as comment, class, dependency, package, usage, generalization, association, attribute and instance specification) and stereotypes. Therefore, the business model is modeled by class diagram which uses these UML standard elements. In target WSMO meta-model, the WSMO ontology, which composes of relation, instance, concept and axiom, is generated. For transformation, two areas of modelling are defined including MDE and WSMO spaces. In MDE space, engineering model for transformation between UML and WSMO and vice versa has defined. This space is split into three layers itself. In M1 layer, four components such as UML 2.0 source model, WSMO UML profile, WSMO/WSML projector and WSMO ontology model are existed. In M2, three meta-models including UML, WSMO and ALT are located. In M3 layer, meta-model is found by using ECORE language. However, transformation between WSMO ontology and UML profile require transformation rules. Three mapping rules have defined based on mapping a class UML stereotyped into WSMO concept, mapping UML properties stereotyped into WSMO attributes and mapping UML instance into WSMO instance. For projection the WSMO to WSML transformation, several transformations should be performed to map WSMO meta-model (serialized XMI format) in MDE space into WSML syntax (WSML format) in WSMO space. A part from the main rules, a rule is defined by translating the WSMO concept in ECORE format into WSML syntax.

XML Link Language (XLink) is mainly focused on formant and syntax without considering semantics and knowledge (Hsu *et al.*, 2014). Moreover, the lack of conceptual modeling in XLink has limited this approach. On the other hand, XLink has metadata association which can provide knowledge representation. A recent MDA with UML profile (such as stereotypes, constraints and tagged value) approach is developed to model XLink application conceptually and visually for different domains (Hsu *et al.*, 2014). To achieve this aim, an OWL-based Language (OWL-L) has been developed for XLink. To make web resources computer-interpretable, two OWL-L links (such as link profile and link model) have described. Agent's information for discovering the link is provided by link profile, while information about exploiting the link (by agent) is provided by link model. Currently, XLink needs software engineering approach like UML for visualization. However, UML cannot provide concrete domain for specific concepts like XLink. Therefore, XLink framework is modeled in CIM level to capture the XLink's properties and concepts. In addition, UML profile is designed to model resources link relationships visually in PIM level. Then, this UML profile model in PIM level is transformed to PSM level by XSLT. Lastly, various XLink documents are mapped into XML format automatically for different application domains. First, resource link relationship is modeled visually by adapting XLinkUML profile through UML tools. Second, UML tools exports XMI document which is imported by U2XLink Transcoder. Third, XMI2XLink and XMI2OWL-L style sheets have transformed XMI documents into (a) XLink schema and (b) OWL-L ontology document respectively. As seen in Figure, UML XLink profile is exported to XMI file which contains the XLink model. This XMI file can be retrieved and parsed. Furthermore, this XMI file may be transformed into OWL-L ontology or XLink schema by using XSLT. The two main components in this approach are including U2XLink Transcoder and XLinkUML profile.

It seems not only the proposed approach generates relationships between web service's resources but also it can adopt software engineering techniques for improving the software quality in XML development.

Meta-model based: To design and develop SWS, independent and general meta-model (UML profile) has used to represent atomic and composite processes. The four main approaches in this category are discussed in details as following:

In this approach, SWS description is modeled by UML profile which is created based on UML activity model elements meta-model and UML Ontology Profile (UOP) meta-model (Grønmo and Oldevik, 2005). A new reciprocal tool called UML Model Transformation (UMT) is applied to map UML diagrams into OWL-S specifications. For this mapping, XSLT script has

applied. Composition modeling, discovery and selection are three main steps in this approach. Firstly, a UML web service model is created to represent task, task name, input, output, task control and dataflow among the tasks. OCL has been used in composition modeling step to represent the conditions which causes annotate UML model semantically. Secondly, discovery step is done based on match making algorithm among services' capabilities automatically. This step required to assume availability of service interface description with location and semantic description in web service registry. Therefore, lexical document is derived from composition model which has been created in first step, to parse by semantic match making algorithm on inputs, outputs and categories. A set of candidate service for each task is the output of this step. Thirdly, the designed task in UML model is selected from the appropriate services. For this purpose, reverse engineering is taken place for SWS description in UML. Selecting appropriate service for each task caused providing concrete composition model which in turn generate SWS description. To develop integrated platform and high-level graphical SWS modeling approach, UML profile can provide enough expressive for reciprocal transformations between UML and OWL-S. The main contribution of this approach is demonstration of UML capability as integration platform for SWS modeling. The UML modeling can provide expressiveness, independence and readability which provide enough independency for SWS developers to be free from using low-level XML coding and detaching from a particular SWS language. UML profile provides an extension on UML meta-model based on three elements such as stereotypes, tagged values and constraints. The properties and relationships of the ontology concepts is captured by UOP which represents semantic concept and defines SWS composition models. Transformation UML to OWL-S (or vice versa) for SWS composition is based on two tasks: provide UML view as reengineering process and annotated composes service semantically.

In this approach, SWS has been described independently by UML profile and meta-model which is defined based on Ontology Definition Meta-model (ODM) to compatible with WSMO, OWL-S, WSDL-S and SWSF (Lautenbacher and Bauer, 2007). An Open Architecture Ware-language Xpand (OAWX) is applied to implement specified transformation rules informally. Then, code is generated from meta-model which constructed from five interactive packages. These packages of ontology are similar to ODM which is including: interface package (for WSDL service and semantic description), Service Provider package (for SWS and non-functional description), functional package (for annotation each single step) and ProcessFlow package (for merging the extended

concepts of ServiceProvider package). Actually, ODM is based on four platforms in PIM including OWL, RDF, topic maps and common logic. Meta-model and UML profiles are defined for RDF and OWL as well as mapping between SWS and meta-model. Therefore, SWS meta-model can model ontologies in form of UML profile based on ODM. Although meta-model cannot integrate with MDA layers perfectly, it has constructed for every specific platform and SWS language. In addition, meta-model can generate the code directly. Although ODM has developed different construct, the most common is generating exist specializations OWL classes. After grounding ontology, service's infrastructure should be modeled by describing operations (input and output) from semantic ontology element. Then, each SWS is processed with specific behavior. However, the lack of SWS rules to integrate the model with developed meta-data is the main limitation of this approach.

Similar to Grønmo *et al.* (2004), in this approach, UOP and UML activity meta-model create UML profile (Bensaber and Malki, 2008). Due to static structure of UML, the capabilities of UML modeling is extended through stereotypes, custom data types and tagged values to model different constructs in OWL-S. However in this approach, logical expressions, pre/post conditions and the effects are not perfectly transformed from UML to OWL-S. Three main steps of this approach are including:

- Reverse engineering process
- Annotation process
- Conversion process

Firstly, reverse engineering manner is done to convert WSDL document to UML profile. For this reason, two activities such as interface modeling (for specifying interface and operation of the service) and workflow modeling (for specifying the order, identifying operation's activity and internal behavior of web service operation) should be performed. By reverse engineering process, two UML diagrams (UML class diagram and UML activity diagram) are generated. JWSDL and DOMSAX APIs process the WSDL document to parse XML schema. Also, transformation rules is applied to generate UML class and activity diagrams which are corresponding to WSDL documents. Secondly, UML model is defined by importing ontologies to represent semantic aspects. To skeleton UML profile, there is possibility to enrich them with UDDI and developer information. Furthermore, some information like category is helped developers to find ontology domain properly which annotate the UML model semantically. Therefore, UML interface can visualize ontology by mapping XSD

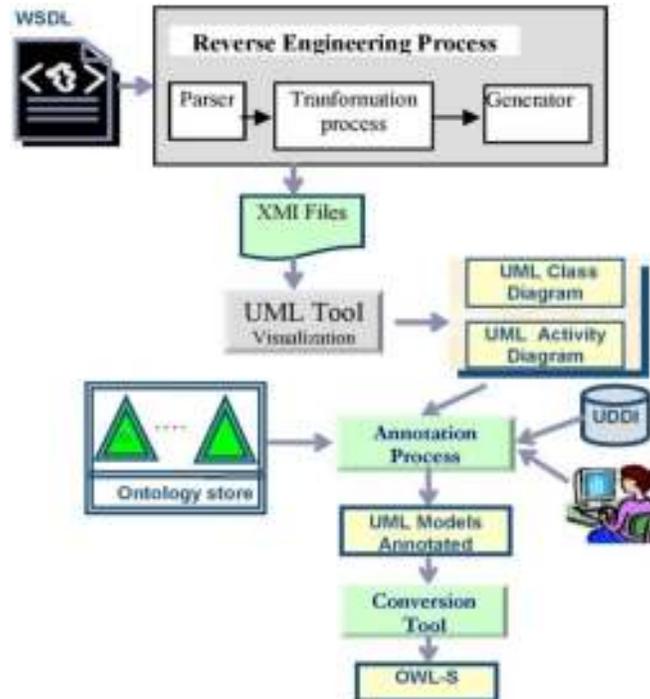


Fig. 2: The overall framework in approach (Bensaber and Malki, 2008)

complex types. Thirdly, the annotated UML model generates OWL-S description by using XSLT. The transformation rules rely on UML profile which completed with a little additional information about developer in second step. By running multiple transformation on input file via conversation tool, various OWL-S documents such as *service*, *servicemodel*, *serviceprofile* and *servicegrounding* is generated. Due to access URIs and bindings, transformation from WSDL to UML (equivalent to transformation from PSM to PIM by reverse engineering) activity diagrams can provide automatic grounding process. Basically, service grounding is consisted from a set of atomic grounding process which is mapped the process type from OWL-S specification to WSDL specification. By using reverse engineering, conversation rules are defined to convert WSDL to UML automatically. In this approach, two conversation rules from WSDL to UML (based on both UML class diagram and UML activity diagram) have studied. Figure 2 summarizes the main steps in this approach.

The overall framework in approach Bensaber and Malki (2008) and Belouadha *et al.* (2010): This approach is provided easy and extensible MDA solution based on atomic and composition of SWS meta-model which independent from any SWS languages. Business service model is used for modeling web services due to realization a number of the interfaces or operations which consist of input, output and fault. The specific layer of SWS uses semantic concept which associate with operations, interface and parameter's type. This

semantically concept is based on ontological concept which may link to category of the service. Furthermore, semantic concept and operation are attached to each other to provide functionality information. Two attributes called *LiftingScheme* and *LoweringScheme* are used for meta-model in order to map between semantic concept and data types of the specific parameter. The proposed UML profile not only provide mapping schemas based on *LiftingScheme* and *LoweringScheme* but also, use tagged value like semantic concept's URI to annotate for data types mapping. Although there is possibility to model web service binding by tagged values, it is not necessary at modeling level due to specification of the service execution in PSM level. The behavior of SWS composition is modeled by SAWSDL and BPMN notations are used to generate BPEL executable file via ATL transformation language. More than one aggregation of atomic or composite services is made semantic composite services. Although it is inherited similar properties with atomic services, it may have specific behavior in terms of orchestration and coordination for those operations. This meta-model can be considered as same level with PIM in MDA approach. Basically, SAWSDL needs PSM meta-model for PIM models which already have created by UML profile. However, WSDL file should be integrated by SAWSDL of the semantic annotation. In addition, a set of meta-class should be comply with WSDL and SAWSDL to construct SAWSDL meta-model. The meta-model in this approach is same as PSM level in

MDA. Behavior of composite web service provides information such as orchestration, operation and coordination of the compose web service. Both modeling of composite web service and business process are same which adapts organization the process execution. For orchestration of business process, BPMN is more convenient than UML activity diagram. Not only BPMN is become standard for business modeling but also, it can map business modeling to executable and immediate code. Both BPML and BPEL4WS can provide specifications for message, business, dataflow, events, transactions, rules and exception. The activity workflow which is known as web service orchestration can constitute the business process.

Formal-based: The formal model is highly desirable for standardization and support of WSMO language (Narock *et al.*, 2014). To better development of WSMO, the semantics and syntax of the WSMO might be modelled formally.

Providing information with natural language (i.e., English) can cause ambiguously, redundancy and contradiction in modeling SWS by WSMO (Wang *et al.*, 2012). Moreover, it is difficult to extend and revise the WSMO description while WSMO is continually evolved. Therefore, the definition of semantic WSMO model with formal model is highly desirable. Furthermore, formal representation of WSMO can improve extensibility and reusability against evolutionary process.

In this approach, a unified framework have been defined precisely for various WSMO aspects based on a formal Object-Z (OZ) which is modeled formally (Wang *et al.*, 2012). Actually, OZ is Z formal specification language which is extended with object oriented to provide a single formal model for the syntax (well-formation of a WSMO model), the static (meaningfulness of a WSMO model) and dynamic (interpretation and execution of a WSMO model) semantics of WSMO. These three separate aspects of the language can be described formally by using OZ in one single and unified framework. OZ not only can improve the consistency among different aspect of a single framework but also, enhance the structure of large specifications. Specifications of the WSMO with OZ have four advantages over other formalisms. First, OZ can adopt with object oriented model style with modularity and reusability. Second, OZ is fully studied the semantics due to its close relation with z standard work. Third, OZ can provide object-oriented constructs to focus on WSDL instead of low level modeling issues which highly simplify formal modeling. Fourth, OZ can model both OWL-S and WSMO formally in order to comparison and identification between both languages.

The different OZ classes model the WSMO elements. Attributes of the OZ class captures the language syntax. The static semantic of the language is captured by class invariant which is defined by predicates. Class operation is captured changing in the web service state which is WSMO dynamic service. In order to formalize the syntax and static semantics, identifiers, WSMO elements and annotations should be modeled. Thus, OZ class ID has been used to denote all identifiers. Also, annotation and WSMO elements are modeled by using OZ class. However, WSMO elements has divided to top level elements and nested elements. Top-level elements themselves have four WSMO elements such as ontologies, web services, goals and mediators. In contrast of WSMO elements that cannot be attached to themselves and other variables, nested elements are attached to WSMO elements. For top-level ontology elements, the formal invariant is provided by defining a set of concepts, instances, relations and some axioms. For top-level web service elements, capability (precondition, postcondition, assumption and effect,), interfaces (orchestration and choreography), non-functional properties, used mediators and imported ontologies have been defined. The top-level goal elements can be modeled formally by just modifying gwMediator and ggMediator from the top-level service elements. For top-level mediator elements, ooMediator has been applied. In order to formalize the dynamic semantics, the formalize model for syntax and static semantics have been extended through values, variables, logical expression and web service execution model. Due to web service is the key element of service in terms of behavior and functionality, the execution of service's semantic has been concentrated. Basically, an atomic piece of functionality is represented a web service which is reused to build the complex piece through formal representation of capability, interface and transition rules. It can be concluded that this approach can provide unambiguous model and it can eliminate documentation's errors which improve future development capability.

Query-based: Unfortunately, SWS annotation process is mainly performed manually which causes error-prone, consuming the time and some difficulties. Although some efforts have been done to automatic this process, these efforts have required ontology building that is a difficult process. Moreover, these efforts have been suffered from matching problems which makes them to face low percentage problem. Query-based SWS allows the developers to annotate their syntactic services more easily and effectively in semi-automatic manner. The query engine can execute the standard query template based on structure and name. Furthermore, an automatic ontology selection mechanism should be developed to minimize the matching computational expensiveness while multiple ontologies are performing the annotation process.

Unfortunately, SWS annotation process is mainly performed manually which causes error-prone, consuming the time and some difficulties. Although some efforts have been done to automatic this process, these efforts have required ontology building that is a difficult process. Moreover, these efforts have been suffered from matching problems which makes them to face low percentage problem. Query-based SWS allows the developers to annotate their syntactic services more easily and effectively in semi-automatic manner. The query engine can execute the standard query template based on structure and name. Furthermore, an automatic ontology selection mechanism should be developed to minimize the matching computational expensiveness while multiple ontologies are performing the annotation process.

Al-Asswad (2011): In this approach, SWS is modeled without applying ontologies. For this aim, WSDL extracts the semantics and data to produce a standard query template. In other word, combination of the service ontologies and WSDL file are fed to build annotated WSDL output file. This approach accepts WSDL file and ontologies as input to extract annotate WSDL file as output. The five major phases in this approach including:

- Concept extraction
- Query filling and concept filtering
- Query execution
- Result assessment
- SAWSDL annotation. Each of these phases has discussed in the details in the following:

Concept extraction: In this phase, a set of output concepts are extracted from a WSDL file automatically to construct the service elements which should be annotated. These extracted concepts are consisted of relations among complex elements and their child, complex and simple types. Extraction of service knowledge from WSDL source are utilized by A Nearly New Information Extraction System (ANNIE) which is consist of four components such as Document Reset, ANNIE Tokenizer, ANNIE Sentence Splitter and JAPE Rules.

Query filing and concept filtering: The extracted concepts in previous phase are fed into concept filtering phase in order to extract a set of query instances. Due to insignificant meanings of some the concepts, the annotation process should be excluded them. Therefore, concept filtering is performed manually because there is no any effective filtering technique. However, query filling creates query instance based on the standard query template for simple and complex types. In fact, the query template is defined a standard format for all of the query instances and it is related to service elements. The query filling phase doesn't need the technical or knowledge domains.

Query execution: In this phase, ontology and a query instance are accepted as inputs to produce a set of query results as the output. Therefore, this phase applies query execution engine to execute query instances against ontology automatically. To implement query execution engine, a novel name-based matching which is using CN-Matching, has designed. In fact, the similarities between Compound Nouns (CNs) and single terms are calculated by CN-Matching. On the other hands, using same label between two ontological classes may have different meaning which can be improved by structural matching which measure similarities between ontological entities.

Result assessment: In this phases, a set of appropriate and inappropriate correspondences are extracted from the set of query results from previous. Then, the results of the matching, which have already done by query execution, are faced to user to verify the correct and wrong ones as appropriate and inappropriate correspondences respectively. Due to involvement of the human to improve the query results accuracy, this phase should be done manually.

SAWSDL annotation: In this phase, the set of appropriate ontological correspondences (in previous phase) extract the annotated WSDL elements automatically by using SAWSDL formant as W3C recommends for SWS description. Therefore, a model reference element (URI) is added to the tag of the candidate service element by automatic annotation which parsing the WSDL file line by line for finding the candidate service element. If the candidate service element is found, then the tag is added to that line. However for inappropriate correspondences, the current ontology should be extended in different existed ontology domains in the repository.

As mentioned, only three phases (such as concept extraction, query execution and SAWSDL annotation) have been performed automatically. Consequently, the main contribution of this approach is claimed on these phases. Figure 3 depicts the overall phases in the proposed query-based approach.

The overall process flow for annotation

The evaluation has been done based on annotated elements such as Percentage, F-Measure, Precision and Recall. This evaluation reveals that the proposed approach were provided complete, accurate and high percentage annotated results by extending ontology mechanisms.

SYNTACTIC APPROACHES

In order to generate BPEL4WS in Mantell (2003), the UML activity diagrams are used. Skogan *et al.* (2004) have proposed a paradigm to design web service composition using activity diagram. The outputs of this approach are BPEL4WS and WorkCo specification.

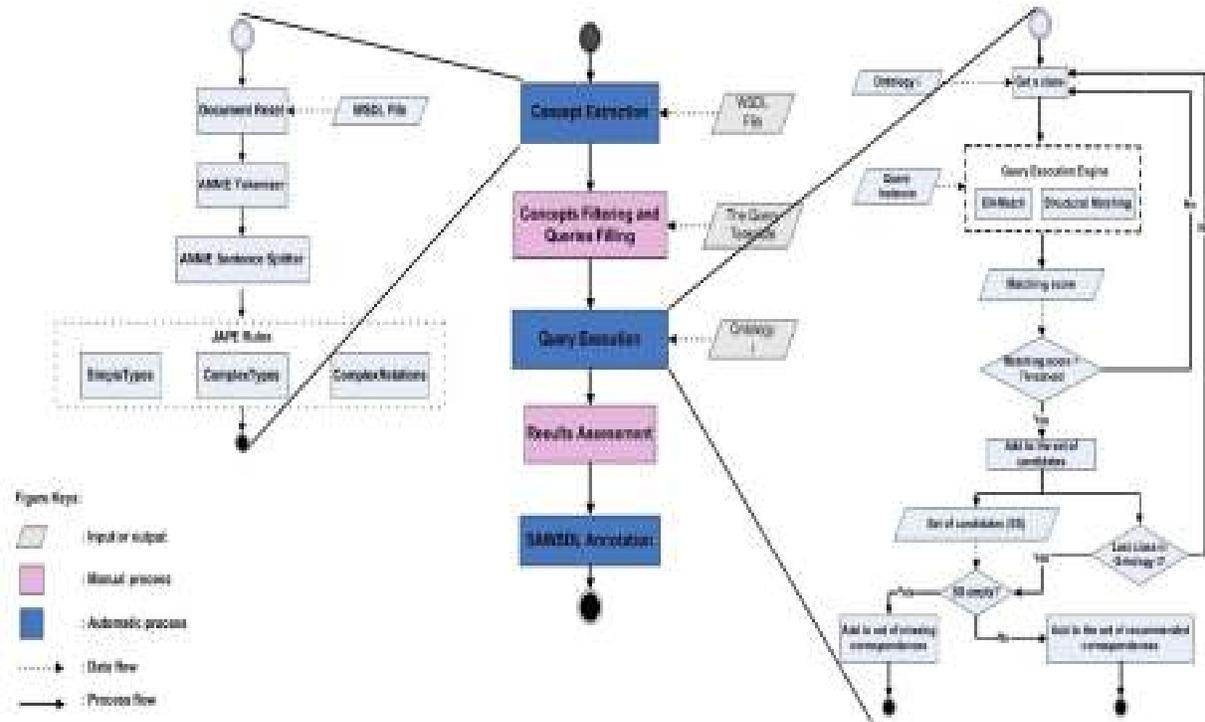


Fig. 3: The overall process flow for annotation

Grønmo *et al.* (2004) have proposed the conversion rules for transforming from UML model to WSDL description. Ortiz and Hernández (2006), Ortiz and Hernandez (2007) and Ortiz and de Prado (2009) have presented a model driven development using UML and Service Component Architecture (SCA) modeling. Ouyang *et al.* (2006) have provided a method that can automatically generate BPEL from BPMN.

Yu *et al.* (2007) have proposed a framework for developing Web service based on model driven which makes a use of CCA choreography specification and the Enterprise Distributed Object Computing (EDOC) CCA specification to generate PIM level. Then, in accord with functional decomposition the PIM can be divided into sub-PIMs that are implemented in Web services. These PIMs are converted into the service interface models and BPEL. Finally, on specific platform Web services are created by the sub EDOC PIMs implementation. The automated transformation between EDOC CCA models and WSDL interface and the transformation between EDOC CCA models and BPEL are the keys of this approach.

Lee (2009) have presented the way to use BPM development process and Model Driven Architecture for developing Web service system. The aim of this approach is to develop a solution for Web service and SOA in which the Web service development implementation supports the system modeling strategy and MDA model. The approach have adopted the

system develop from Alhir (2010) and MDA Web service development from Grønmo *et al.* (2004) to create BPEL and WSDL.

Quintero *et al.* (2010) have developed a model-driven conceptual modeling to tackle the modeling and transformation problem. In addition this approach has provided an automated code generation of WSDL document. Qafmolla and Nguyen (2010) has proposed a MDA based approach in conjunction with tools available in market to develop Web services using Kermeta for describing meta-model transformation. There are some other approaches that tried to generate WSDL and BPEL in MDA manner such as Zhao *et al.* (2009), Martinez *et al.* (2005), Mayer *et al.* (2008), Bezivin *et al.* (2004) and Kath (2004).

Comparison evaluation: In this section, all approaches that are described in section 2.8 are classified and evaluated using a comparative study. In addition to ontology based classification that described in section 2.8 and Figure, these approaches are classified in three more categories in terms of standard properties of model-driven approach. This research adopts and updates the criteria and categories from Brambilla *et al.* (2007) and Timm approaches. Furthermore, the comparative evaluation is performed based on evaluation criteria. In the following, the comparison among the most relevant approaches are presented for modeling language, transformation language, semantic

Table 1: Modeling language used to design Platform Independent Model (PIM)

Approaches	Modeling language									Result
	UML					Non-UML				
	CD	AD	SQD	STD	UP	Condition	BPEL4WS	BPMN	WebML	
Gronmo <i>et al.</i> (2004)	√	√			√	OCL				UML
Yang and Chung (2006)	√			√		GUI	√			NUML
Lautenbacher and Bauer (2007)	√	√			√	constraint				UML
Torres <i>et al.</i> (2006)	√	√		√	√	OWL&S				UML
Acuña and Marcos (2006)	√				√	WSML				UML
Brambilla <i>et al.</i> (2007)								√	√	NUML
Timm (2008)	√	√			√	OCL				UML
Bensaber and Malki (2008)	√	√			√	OCL				UML
	√	√				N/A				UML
Kim and Lee (2009)	√	√	√		√	Constraint				UML
	√				√	OCL				UML
Belouadha <i>et al.</i> (2010)	√				√			√		NUML

CD = Class Diagram, AD = Activity Diagram, SQD = Sequence Diagram, STD = Statechart Diagram, UP = UML Profile

Table 2: Transformation language used to convert one model to another

Approaches	Transformation language				
	XPand	Java	XSLT	Model-to-text	ATL
Gronmo and Oldevik (2005)		√	√		
Yang and Chung (2006)		√			
Lautenbacher and Bauer (2007)	√		√		
Torres <i>et al.</i> (2006)				√	
Acuña and Marcos (2006)	N/A				
Brambilla <i>et al.</i> (2007)			√		
Timm (2008)			√		
Bensaber and Malki (2008)			√		
Kim and Lee (2009)			√		√
	N/A				
Belouadha <i>et al.</i> (2010)					√

web service language and coverage through different criteria such as expressiveness, independence, readability, productivity, reusability and scalability.

Modeling language: Table 1 contains a summary of modeling language used by aforementioned approaches to model Platform Independent Model (PIM) in an MDA based semantic web service development.

The result shows that ten of the thirteen approaches use UML as modeling language and among them almost all of them use Class diagram to represent ontology and atomic process. In addition the majority of them provides UML profile in order to define stereotypes, tagged value, constrains and the like. In order to express conditions most of these approaches employed Object Constrain Language (OCL) which is used to write constraint. Brambilla *et al.* (2007) and Belouadha *et al.* (2010) have used BPMN to model composite processes.

Transformation language: Table 2 shows the most approaches use XSLT which is W3C standard, as their transformation languages. Sanchez D.M *et al.* have proposed a framework based on Acuña and Marcos (2006) framework to represent constraints using OCL. However both approaches did not specified any

transformation model. Indeed, the result shows that the latest efforts to provide a model driven semantic web service development desire to use ATL as transformation language.

Semantic web service language: The approaches are classified in terms of semantic web service language in Table 3. For the sake of clarity approaches are highlighted to specify the classification result. The result shows the most model-driven semantic web service developments that have been done are based on OWL/OWL-S. However, some of them claim that their approach can support or generate a semantic description base on WSMO. According to the study, only three of them are capable to generate WSMO based description. For example in Gronmo *et al.* approach, there are no equivalents for the goal and the mediator concepts. Although, Lautebacher and Bauer have provided metamodel that can specify almost all languages. There is no sample or transaction rules for WSMO and they only provided an example for OWL-S. Until this moment, the only approach that can be considered as a model driven approach to generate WSMO based semantic description is Brambilla *et al.* Indeed, Acuña and Marcos (2006) and Sanchez D.M *et al.* have proposed

Table 3: The targeted semantic web service language/ontology to provide Platform Specification Model (PSM)

Approaches	Semantic web service language/Ontology				
	OWL/OWL-S	WSML/WSMO	SWSL/SWSO	WSDL-S	SAWSDL
Grønmo and Oldevik (2005)	√	√: low	×	×	×
Yang and Chung (2006)	√	×	×	×	×
Lautenbacher and Bauer (2007)	√	√	√	√	×
Torres <i>et al.</i> (2006)	√	×	×	×	×
Acuña and Marcos (2006)	×	√	×	×	×
Brambilla <i>et al.</i> (2007)	×	√	×	×	×
Timm (2008)	√	×	×	×	×
Bensaber and Malki (2008)	√	×	×	×	×
Kim and Lee (2009)	√	×	×	×	×
Belouadha <i>et al.</i> (2010)	×	√	×	×	×
	×	×	×	×	√

Table 4: The coverage of approach based on requirements

Approaches	Coverage/Support					
	Composition	Execution	Grounding	Existing WS	Software Methodology	Meta Model
Grønmo and Oldevik (2005)	√	×	×	√	UF	√
Yang and Chung (2006)	√	√	√	√	UF	×
Lautenbacher and Bauer (2007)	√	N/A	N/A	×	UF	×
Torres <i>et al.</i> (2006)	×	×	×	×	SEM	×
Acuña and Marcos (2006)	√	×	×	×	SEM	√
Brambilla <i>et al.</i> (2007)	√	√	√	√	SEM	×
Timm (2008)	√	√	√	√	UF	×
Bensaber and Malki (2008)	√	√	√	√	UF	√
Kim and Lee (2009)	√	×	×	√	UF	√
	√	×	×	×	UF	×
Belouadha <i>et al.</i> (2010)	√	×	×	×	SEM	√
	√	√	√	√	UF	√

SEM = Software Engineering Method,, UF = UML Formalism

approaches based on WSMO but, their lack of transformation rules make it uneasy to implement and evaluate. In addition, they just focused on the part of framework which is to represent PSM. So far, there is only one approach which tried to develop semantic web service based on SAWSDL.

Coverage: Some requirements and concepts must be considered through semantic web service development. These requirements contain supporting composition, invocation, grounding and existing Web services usage. It is important to develop a semantic web service during its software development life cycle using an adequate software engineering method. Table 4 illustrates that except Torres *et al.* (2006) the rest of approaches support atomic and composite process. Due to the existing of the large amount of Web services available in internet, most of approaches rather to use them during the modeling and grounding. Creating appropriate meta-model helps to provide an independent model to support several semantic web service languages as Lautenbacher and Bauer have proposed. Indeed, these approaches can be divided into two different categories: 1) software engineering method and 2) UML formalism. Almost all WSMO based approaches are place in first category. Brambilla *et al.* (2007) have used WebML as a methodology and

the rest have used MIDAS framework as Web Information System (WIS).

Comparison criteria: In this part, a set of comparison criteria is used based on above classification and summary in order to evaluate related approaches.

Expressiveness: Approaches may use various modeling languages to specify the description of semantic web services. The models should have sufficient semantic annotations to provide a complete semantic description. Expressiveness refers to the ability for transforming a model to and from a complete semantic web service document. An expressive model should support atomic and composite process along with their conditions (including pre and post-condition). Evaluation of this expressiveness criterion is given as following:

- **High:** The approach supports sufficient semantic annotation in the model design and generates the equivalent semantic web service description.
- **Medium:** The approaches cannot fully support semantic annotation in the model design and generates the equivalent semantic web service description.

Table 5: Evaluation on expressiveness criterion

Approaches	*Semantic annotation	*SWS generation	Expressiveness
Grønmo and Oldevik (2005)	Middle	Middle	Middle
	Middle	Middle	Middle
Yang and Chung (2006)	High	Middle+	High
Lautenbacher and Bauer (2007)	Middle	Low	Low
Torres <i>et al.</i> (2006)	Middle-	Middle	Middle -
Acuña and Marcos (2006)	Middle -	low	Low
Brambilla <i>et al.</i> (2007)	Middle	High	Middle +
Timm (2008)	High	High	High
Bensaber and Malki (2008)	High	High	High
	Middle	Middle -	Middle -
Kim and Lee (2009)	High	Middle	Middle +
	Middle	low	Low
Belouadha <i>et al.</i> (2010)	Middle +	High	High

*- High: fully support, Middle: partly support, low: do not support

- **Low:** The approach does not support generation of equivalent semantic web service description.

Timm (2008) have proposed a high level expressiveness approach due to combination of the class, activity diagram along with an UML profile. The BPEL4WS elements used in approach are expressive enough to represent semantic web service. However the condition should be written manually. Lautenbacher and Bauer approach cannot provide the translation of conditions specification into the SWRL. This limitation causes the expressiveness of this framework decrease. In addition, the Torres *et al.* approach has low degree of expressiveness because; the service profile should be created manually. In Brambilla *et al.* (2007) developer should provide additional input to create a complete semantic web service specification based on WSMO. Indeed, Grønmo *et al.* (2004) cannot provide the specification of all control construct in the OWL-S. Thus, it cannot provide complete semantic web service description. The two approaches that have worked on MIDAS-S are not expressive yet since there is no equivalent of WSMO specification for their models. The result is shown in Table 5.

Independence: In the concept of Model Driven Architecture, the model must be completely independent of the out coming language so that they can be used in different purposes. In the development of semantic web service based on MDA, it is important to use the same set of models to generate different semantic web service languages as out coming languages. The independence and reusability both are the critical keys in MDA. Approach high level independence due to use of BPEL4WS as PIM. In Lautenbacher and Bauer approach several semantic web service standards are addressed in a meta-model which makes it more independent than other approaches. Torres *et al.* (2006) has no independency from the outgoing semantic web service language. Because, the approach uses OWL and OWL-S syntaxes to express constraints in state charts. Kim and Lee approach has no independency from semantic web service languages. Their approach just supports OWL-S specification and control constructs have to be specified by SWRL in the

UML model. The Brambilla *et al.* (2007) method has been proposed based on WSMO and it has a low degree of independency. The Grønmo *et al.* approach uses a meta-model for its UML profile. However, it cannot support WSMO completely as output. Like Acuña and Malki approach, the method has no independency from outgoing language and both have been intended based on WSMO.

Readability: Readability refers to the mean that an approach is proposed in the way of being easy to understand, specified and interpreted for skilled modelers so that they can use them to create complex semantic web service specification. Readability is also very important wherein the model may be used for design communication and maintenance. An approach must be readable to both human for convenient study and use and to the computer for automating process. The degree of readability in Timm approach is considered as high due to the simplicity of their visualization so that they only display the most important attributes. However it is considered as low level in due to the lack of visual notation. The diagrams of the class that have been created for using Lautenbacher and Bauer approach, is readable and they bear all of the information that is related to the modeler with clarity in high degree. However, the activity diagram become hard to read, because this approach captures output, input, effects and preconditions and make the diagram more complex. The degree of readability in Torres *et al.* (2006) is considered as high due to use of readable state-chart diagrams. Like Acuña and Malki approach, the method has a high degree of readability due to its simplicity and MOF based modeling.

Productivity: Productivity in this research refers to the availability of automatic semantic annotations and code generation. Evaluation of this productivity criterion is given as following:

- **High:** The approach can automatically generate the complete semantic web service description and generate the executable code.

Table 6: The scalability criterion

Approaches	Composition	Execution	Grounding	Existing WS	Complex processes	Scalability
Grønmo and Oldevik (2005)	√	×	×	√	×	Middle
	√	√	√	√	×	High
Yang and Chung (2006)	√	N/A	N/A	×	×	Low
Lautenbacher and Bauer (2007)	√	√	√	√	×	High
Torres <i>et al.</i> (2006)	×	×	×	×	×	Low
Acuña and Marcos (2006)	√	×	×	×	×	Low
Brambilla <i>et al.</i> (2007)	√	√	√	√	√	High
Timm (2008)	√	√	√	√	×	High
Bensaber and Malki (2008)	√	√	√	√	×	High
	√	×	×	√	×	Middle
Kim and Lee (2009)	√	×	×	×	√	Middle
	√	×	×	×	×	Low
Belouadha <i>et al.</i> (2010)	√	√	√	√	N/A	High

- **Medium:** The approach can semi-automatically generate the semantic web service description or part of them and generate the executable code.
- **Low:** The approach does not support generation of equivalent semantic web service description.

With respect to the above criteria, Brambilla *et al.* (2007) provide an approach with high degree of productivity. Because, they have adopted WebRatio that is able to automatically generate J2EE code of an application designed by WebML. In addition, WebML diagrams are also automatically generated via business process specification. Similarly, Timm (2008) provide a tool (SPEX) to automatically generate semantic web service specification. There is no direct automatic transformation in approach. The Kim and Lee method automatically generates an OWL-S ontology with no service grounding. There is no automated transformation for logic expressions in the Grønmo *et al.* method. Yang and Chung (2006) have proposed an automatic translation to OWL-S specification.

Reusability: The reusability of a model depends on the independency of a model. An approach has the high degree of reusability when it's PIM which is provided by modeling language is completely independent of outgoing language. In order to reuse the Timm model for different semantic web service language, a new UML profile should be provided. This lack of independency makes this approach become un-reusable. However, it is provides a reusable service grounding. This research considers as low level approach in terms of reusability. Because the model is not enough readable and it makes difficult to understand and reuse the same model for another purpose. The models developed via Lautenbacher and Bauer approach have a high degree of reusability due to its independency.

Scalability: The scalability in this research refers to the ability for approach to perform when scaling to big application. It must be evaluated whether the approach is able to use for developing semantic web service composition having complex activities and processes. It also refers to the coverage of the approach as discussed in section 2.9.4. Brambilla *et al.* (2007) approach has

the high level of degree in scalability due to use of J2EE, WSMX and Jakarta Struts. In Timm approach, the activity diagrams are too simple and they cannot support complex diagrams. However, the coverage of their approach is high. The scalability of is considered as high level due to the use of BPEL4WS and OWL-S API. In Kim and Lee approach the activity diagrams can be more complicated when a composite process has more than four activities due to use of constraints for modeling conditions. The Grønmo *et al.* approach does not support complex UML diagrams and it does not support WSMO completely. Much like most of the approaches Yang and Chung method cannot support complex composition. The result is shown in Table 6.

Comparison of approaches: Table 7 shows the comparative evaluation of model-driven semantic web service development approaches based on the aforementioned criteria. There are several facts that can be extracted from this comparative analysis. First, the approaches that have used meta-model in their UML profile have the high degree of independency and reusability. However the result shows where the meta-model is not used, the expressiveness get an acceptable degree. Second, all framework proposed based on WSMO have tried to follow the software engineering methodology rather than UML formalism to extract the semantic descriptions. The result shows that they have poor independency and reusability in the model design.

Third, the UML based semantic web service development approaches have admissible degree of productivity. Indeed, as a result of this comparison the Brambilla *et al.* (2007), Bensaber and Malki (2008) and Belouadha *et al.* (2010) approaches are the best in developing semantic web service based on MDA. However, they are taking advantage of different semantic web services languages as an output. Their degree of coverage, scalability and productivity are high. Another result that can be considered is the latest approaches tried to use ATL as transformation language. As a conclusion of this comparison evaluation, there is no complete approach that can meet the model driven architecture requirement

Table 7: Comparison criteria

Approaches	Comparison Criteria					
	Expressiveness	Independency	Readability	Productivity	Reusability	Scalability
Gronmo and Oldevik (2005)	Middle	√	√	Low	√	Middle
	Middle	√	×	Low	√	High
Yang and Chung (2006)	High	√	×	High	√	Low
Lautenbacher and Bauer (2007)	Low	√	√	Middle	√	High
Torres <i>et al.</i> (2006)	Middle -	×	√	Low	×	Low
Acuña and Marcos (2006)	Low	×	√	Low	×	Low
Brambilla <i>et al.</i> (2007)	Middle +	×	√	High	×	High
Timm (2008)	High	×	√	High	×	High
Bensaber and Malki (2008)	High	√	×	High	√	High
	Middle -	√	×	Middle	√	Middle
Kim and Lee (2009)	Middle +	×	√	Middle	×	Middle
	Low	×	√	High	×	Low
Belouadha <i>et al.</i> (2010)	High	√	√	Middle	√	High

yet. Considering the advantage of each approach and providing an adequate solution to integrate the sufficient parts of aforementioned approaches in one comprehensive framework can meet all the requirements.

CONCLUSION

And finally, majority of current approaches are only describing semantic web service in the form of UML diagrams while the semantic description of services should be occur from the life cycle of developing services (Brambilla *et al.*, 2009). Regarding to the current approaches, there is no UML based independent framework to automatically generate semantic web service description during application development life cycle.

REFERENCES

Acuña, C.J. and E. Marcos, 2006. Modeling semantic web services: A case study. Proceeding of the 6th International Conference on Web Engineering (ICWE, 2006). Palo Alto, CA, United States.

Al-Asswad, M.M., 2011. Semantic information systems engineering: A query-based approach for semi-automatic annotation of web services. Ph.D. Thesis, School of Information Systems, Computing and Mathematics, Brunel University.

Alhir, S.S., 2010. Understanding the Model Driven Architecture (MDA). Retrieved from: <http://www.methodsandtools.com/archive/archive.php?Id=5>. (Accessed on: August 20, 2010)

Belouadha, F.Z., H. Omrana and O. Roudies, 2010. A model-driven approach for composing SAWSDL semantic web services. *Int. J. Comput. Sci. Issue*, 7(2).

Bendraou, R., J.M. Jezéquel and F. Fleurey, 2009. Combining Aspect and Model-Driven Engineering Approaches for Software Process Modeling and Execution. In: Wang, Q. (Ed.), *Trustworthy Software Development Processes*. Lecture Notes in Computer Science, Springer-Verlag, Berlin, Heidelberg, 5543: 148-160.

Bensaber, D.A. and M. Malki, 2008. Development of semantic web services: Model driven approach. *Proceeding of the 8th International Conference on New Technologies in Distributed Systems (NOTERE '08)*. Lyon, France.

Bensaber, D.A. and M. Malki, 2012. Model driven approach for specifying WSMO ontology. *Proceeding of the ICWIT*, pp: 203-213.

Berners-Lee, T., J. Hendler and O. Lassila, 2001. The Semantic Web. *Scientific American*, pp: 29-37.

Bezivin, J., S. Hammoudi, D. Lopes and F. Jouault, 2004. Applying MDA approach for Web service platform. *Proceeding of the 8th IEEE International Enterprise Distributed Object Computing Conference (EDOC, 2004)*, pp: 58-70.

Brambilla, M., I. Celino, S. Ceri, D. Cerizza, E.D. Valle and F.M. Facca, 2006. A Software Engineering Approach to Design and Development of Semantic Web Service Applications. In: Cruz, I. *et al.* (Eds.), *The Semantic Web - ISWC 2006*. Lecture Notes in Computer Science, Springer-Verlag, Berlin Heidelberg, 4273: 172-186.

Brambilla, M., S. Ceri, F.M. Facca, I. Celino, D. Cerizza and E.D. Valle, 2007. Model-driven design and development of semantic Web service applications. *ACM T. Internet Techn.*, 8(1).

Brambilla, M., M. Dosmi and P. Fraternali, 2009. Model-driven engineering of service orchestrations. *Proceeding of the World Conference on Services - I*. Los Angeles, CA, pp: 562-569.

Cáceres, P., V. de Castro and E. Marcos, 2005. Navigation Modelling From a User Services Oriented Approach. In: Yakhno, T. (Ed.), *Advances in Information Systems*. Lecture Notes in Computer Science, Springer, Berlin, Heidelberg, 3261: 150-160.

De Castro, V., E. Marcos and M.L. Sanz, 2006. A model driven method for service composition modelling: A case study. *Int. J. Web Eng. Technol.*, 2(4): 335-353.

Fensel, D., F.M. Facca, E. Simperl and I. Toma, 2011. Semantic Web. In: *Semantic Web Services*. Springer, Berlin, Heidelberg, pp: 87-104.

- Grønmo, R. and J. Oldevik, 2005. An empirical study of the UML model transformation tool (UMT). Proceeding of the 1st International Conference on Interoperability of Enterprise Software and Applications. Geneva, Switzerland.
- Grønmo, R., D. Skogan, I. Solheim and J. Oldevik, 2004. Model-driven Web services development. Proceeding of the IEEE International Conference on e-Technology, e-Commerce and e-Service (EEE '04), pp: 42-45.
- Group, O.M., 2010. OMG Unified Modeling Language (UML). Retrieved from: <http://www.omg.org/spec/UML/2.3/>. (Accessed on: September 21, 2010)
- Hsu, I.C., D.H. Ting and N.L. Hsueh, 2014. MDA-based visual modeling approach for resources link relationships using UML profile. *Comp. Stand. Inter.*, 36(3): 648-656.
- Kath, O., 2004. Towards executable models: Transforming EDOC behavior models to CORBA and BPEL. Proceeding of the 8th IEEE International Enterprise Distributed Object Computing Conference (EDOC, 2004), pp: 267-274.
- Kerrigan, M., 2006. WSMOViz: An ontology visualization approach for WSMO. Proceeding of the 10th International Conference on Information Visualization (IV, 2006). London, England, pp: 411-416.
- Kim, I.W. and K.H. Lee, 2009. A model-driven approach for describing semantic Web services: From UML to OWL-S. *IEEE T. Syst. Man Cy. C*, 39(6): 637-646.
- Lautenbacher, F. and B. Bauer, 2007. Creating a meta-model for semantic web service standards. Proceeding of the 3rd International Conference on Web Information Systems and Technologies (WEBIST) - Web Interfaces and Applications. Barcelona, Spain, pp: 376-381.
- Lee, M.C., 2009. Semantic model-driven approach of web service systems. Proceeding of the International Conference on New Trends in Information and Service Science (NISS '09). Beijing, China, pp: 427-432.
- Mantell, K., 2003. Model Driven Architecture in a Web Services World: From UML to BPEL. Retrieved from: <https://www.ibm.com/developerworks/webservices/library/ws-uml2bpel/>. (Accessed on: March 18, 2010)
- Margaria, T., C. Winkler, C. Kubczak and B. Steffen, 2007. The SWS mediator with WEBML/WEBRATIO and JABC/JETI: A comparison. Proceeding of the 9th International Conference on Enterprise Information Systems. Funchal, Madeira, Portugal, Volume SAIC, pp: 422-429.
- Martinez, A., M. Patino-Martinez, R. Jimenez-Peris and F. Perez-Sorrosal, 2005. ZenFlow: A visual web service composition tool for BPEL4WS. Proceeding of the IEEE Symposium on Visual Languages and Human-Centric Computing, pp: 181-188.
- Mayer, P., A. Schroeder and N. Koch, 2008. A model-driven approach to service orchestration. Proceeding of the IEEE International Conference on Services Computing (SCC '08). Honolulu, HI, United States, pp: 533-536.
- McIlraith, S.A., T.C. Son and H. Zeng, 2001. Semantic web services. *IEEE Intell. Syst.*, 16(2): 46-53.
- McIlraith, S.A. and D.L. Martin, 2003. Bringing semantics to web services. *IEEE Intell. Syst.*, 18(1): 90-93.
- Miller, J. and J. Mukerji, 2003. MDA Guide Version 1.0.1. Retrieved from: http://www.omg.org/news/meetings/workshops/UML_2003_Manual/00-2_MDA_Guide_v1.0.1.pdf. (Accessed on: January, 15, 2010)
- Narock, T., V. Yoon and S. March, 2014. A provenance-based approach to semantic web service description and discovery. *Decis. Support Syst.*, 64: 90-99.
- Ortiz, G. and J. Hernández, 2006. Service-Oriented Model-Driven Development: Filling the Extra-Functional Property Gap. In: Dan, A. and W. Lamersdorf (Eds.), *Service-Oriented Computing - ICSOC 2006. Lecture Notes in Computer Science*, Springer-Verlag, Berlin, Heidelberg, 4294: 471-476.
- Ortiz, G. and J. Hernandez, 2007. A case study on integrating extra-functional properties in Web service model-driven development. Proceeding of the 2nd International Conference on Internet and Web Applications and Services, (ICIW '07), Morne.
- Ortiz, G. and A.G. de Prado, 2009. Adapting Web Services for multiple devices: A model-driven, aspect-oriented approach. Proceeding of the World Conference on Services-I. Los Angeles, CA, pp: 754-761.
- Ouyang, C., M. Dumas, A.H.M. Ter Hofstede and W.M.P. Van Der Aalst, 2006. From BPMN process models to BPEL Web services. Proceeding of the International Conference on Web Services (ICWS '06). Chicago, IL, United States, pp: 285-292.
- Paolucci, M., N. Srinivasan, K. Sycara and T. Nishimura, 2003. Towards a semantic choreography of web services: From WSDL to DAML-S. Proceeding of the 1st International Conference on Web Services (ICWS'03), pp: 22-26.
- Qafimolla, X. and V.C. Nguyen, 2010. Automation of web services development using model driven techniques. Proceeding of the 2nd International Conference on Computer and Automation Engineering (ICCAE, 2010). Singapore, pp: 190-194.

- Quintero, R., L. Zepeda and L. Vega, 2010. Model-driven software development of applications based on web services. *Int. J. Web Grid Serv.*, 6(3): 313-330.
- Skogan, D., R. Groenmo and I. Solheim, 2004. Web service composition in UML. *Proceeding of the 8th IEEE International Enterprise Distributed Object Computing Conference (EDOC, 2004)*.
- Timm, J.T.E., 2008. A model-driven framework for the specification, grounding, and execution of semantic web services. Ph.D. Thesis, Arizona State University, Tempe, AZ, USA, pp: 170.
- Torres, V., V. Pelechano and O. Pastor, 2006. Building Semantic Web Services Based on a Model Driven Web Engineering Method. In: Roddick, J.F. *et al.* (Eds.), *Advances in Conceptual Modeling - Theory and Practice*. Springer-Verlag, Berlin, Heidelberg, 4231: 173-182.
- Wang, H.H., N. Gibbins, T.R. Payne and D. Redavid, 2012. A formal model of the Semantic Web Service Ontology (WSMO). *Inform. Syst.*, 37(1): 33-60.
- WSMO Studio, 2009. Retrieved from: <http://www.wsmostudio.org/>. (Accessed on: 18 May 18, 2010)
- Yang, J.H. and I.J. Chung, 2006. Automatic Generation of Service Ontology from UML Diagrams for Semantic Web Services. In: Mizoguchi, R., Z. Shi and F. Giunchiglia (Eds.), *The Semantic Web – ASWC 2006*. Springer-Verlag, Berlin, Heidelberg, 4185: 523-529.
- Yu, X., Y. Zhang, T. Zhang, L. Wang, J. Hu, J.H. Zhao and X. Li, 2007. A model-driven development framework for enterprise Web services. *Inform. Syst. Front.*, 9(4): 391-409.
- Zhao, C., Z. Duan and M. Zhang, 2009. A model-driven approach for dynamic web service composition. *Proceeding of the WRI World Congress on Software Engineering (WCSE '09)*. Xiamen, China, 4: 273-277.