Service Oriented Computing and Applications (SOCA) 2013
The final publication is available at link.springer.com:
http://link.springer.com/article/10.1007%2Fs11761-013-0138-2

DOI 10.1007/s11761-013-0138-2

Integration of Business Process Modeling and Web Services: A Survey

Katarina Grolinger, Miriam A. M. Capretz[1]
Department of Electrical and Computer Engineering, Faculty of Engineering
Western University
London ON Canada N6A 5B9
Phone: 1-519-661-2111 ext. 85478
Fax: 1-519-850-2436
kgroling@uwo.ca, mcapretz@uwo.ca

Americo Cunha
Faculty of Business, Sheridan Institute of Technology & Advanced Learning
Mississauga ON Canada L5B0G5
americo.cunha@sheridanc.on.ca

Said Tazi
CNRS, LAAS
7 Avenue du Colonel Roche, F-31400 Toulouse, France
Université de Toulouse
UT1 Capitole, LAAS, F-31000 Toulouse, France

Abstract: A significant challenge in business process automation involves bridging the gap between business process representations and Web service technologies that implement business activities. We are interested in business process representations such as BPMN (Business Process Modeling Notation) and EPCs (Event-Driven Process Chains). Web Service technologies include protocols such as SOAP (Simple Object Access Protocol), architectures such as RESTful (REpresentational State Transfer) or semantic description languages and formalisms such as OWL-S (Web Ontology Language for Services) and WSMO (Web Service Modeling Ontology). This paper reviews previous work on the integration of business process representations and Web service technologies. It provides a perspective on the field by summarizing, organizing, and classifying the proposed approaches. Consequently, this study has identified opportunities for future research in the field, including the need for a generic transformation approach among arbitrary models, the need to represent mappings in a formalized way, and the necessity of a common execution framework.

Keywords: Business process modeling, semantic Web services, model transformations, ontology, Web-based services

[1] Corresponding Author – email address: mcapretz@uwo.ca

1. INTRODUCTION

A major challenge in the field of business process management (BPM) involves bridging the gap between a business view of the processes as represented through workflow models and an executable view of the processes as represented in the form of Web services that implement business process activities [1-3].

BPM, which is concerned with a business view of the processes, strives to understand, manage, and improve organization business processes with the ultimate goal of meeting clients' needs in the most efficient way. Business process management is supported by a variety of theories, standards, languages, and notations. Theories include Petri net [4] and Pi-calculus [5], while technical aspects include Business Process Modeling Language (BPML) [6], Business Process Modeling Notation (BPMN) [7], Event-Driven Process Chains (EPC) [8], and Yet Another Workflow Language (YAWL) [9].

On the other hand, the executable view of business processes is related to services, which provide a way of executing business processes. In particular, services supply functions of the business processes and can be aggregated to provide complex business process solutions. However, while BPMN, BPML, EPC, and YAWL are used in business process management, services use different technologies. Web services, the most common way of implementing services, use a variety of approaches, methodologies, and technologies for describing, locating, and invoking services, including Representational State Transfer (REST) [10,11], Web Application Description Language (WADL) [12], Business Process Execution Language (BPEL) [13,14], Web Service Description Language (WSDL) [15], and Simple Object Access Protocol (SOAP) [16].

Furthermore, the integration of business and executable process views can be complex due to the need to formalize the semantics of the business process models as well as of the Web services. Traditionally, business processes have been modeled without restrictions on terminology, and therefore, a single business process could be modeled differently by various business experts using diverse terms for the same concept or the same terms for different concepts. This flexible use of terms created difficulties in understanding and exchanging business process models, and presented challenges in using Web services for the execution of those models.

In traditional Web services, i.e., non-semantic Web services, available operations and exchanged messages are described at a syntactic level; the semantic meaning of the data is not specified. As a result, service composition often requires human involvement. However, capabilities of services should cover semantic description in order to achieve wide-scale interoperability. Moreover, the dynamic nature of business interactions, in which new interactions emerge and existing ones change repeatedly, demands a dynamic and efficient automated service discovery and composition. Semantic technologies incorporate semantics with traditional Web services, thus facilitating service discovery, composition, and orchestration [17,18]. Ontologies [19] formally represent knowledge as a set of concepts and their relations; therefore they can be used for incorporating formalized semantics into business processes and Web service models. Specifically, business processes and Web services can be semantically described using ontologies.

This study focuses on the model transformation aspect of business process modeling and Web services integration, while the semantic aspect is explored from the perspective of its role in the transformation.

The term *transformation* is used here to refer to the process of converting data from a source system format into a target system format. It consists of two parts: the first part is the *mapping* which establishes relations between the elements of the source system to its related elements in the target system, while the second part executes the transformation process and creates the target system representation.

Specifically, this paper surveys transformation approaches between business process models and Web services representations with the following objectives:

1) To integrate the work in the field and to provide a perspective on the domain by summarizing, organizing, and categorizing transformations between different business process models and Web service representations. Such a classification will help practitioners locate studies relevant to a specific problem, while a perspective on the domain will help researchers identify future research directions. Because heterogeneity of representations is a major obstacle in transformation, transformations between models of the same category, between two business process models, and two Web service representations are also included.

2) To identify research challenges and opportunities in the field of semantic integration of business and executable process views. The identified opportunities will promote future research in the field.

Given that semantic heterogeneity is part of the integration challenge, the use of ontologies is explored in the paper from two perspectives: first, as a way of incorporating formalized semantics into a business process view, and second, as a facilitator of transformation between two representation models.

Similar to our study, the surveys by Dustdar and Schreiner[20] and Rao and Su [21] focused on Web services as a way of implementing business applications. While the two surveys [20,21] concentrated on the executable view of the processes, specifically on the Web service composition; our study examines the transformation aspect of business process modeling and Web services integration. The business view of the processes is the subject of the survey conducted by van der Aalst *et al.* [22]. Nevertheless, like our study, van der Aalst *et al.*'s work indicates the importance of formal semantics for the unambiguous process descriptions and highlights the significance of model verification and analysis. However, van der Aalst *et al*.'s survey is concerned with the business view of the processes, while our study examines the integration of business process modeling and Web services.

The workshop on XML Integration and Transformation for Business Process Management [23] aimed to investigate the role of service computing, semantic technologies and XML transformations in relationship to BPM. Similar to our work, this workshop addressed the issues of integration and transformation; however the workshop focused only on XML approaches, whereas our work aims to address transformations regardless of the approach applied.

The remainder of this paper is organized as follows: Section 2 describes the review process, while Section 3 introduces basic concepts associated with business process modeling and Web service representations. The various transformations approaches between representations are reviewed in Section 4, and ontologies for business process modeling are considered in Section 5. The challenges and opportunities identified in this study are described in Section 6, while Section 7 provides conclusions. An appendix has been added to the paper containing a list of acronyms and abbreviations.

2. Review Process

2.1 Paper inclusion criteria

The main criterion for including a paper in this review is that the paper addresses the integration of business process modeling and Web services representations. Specifically, there are two categories of relevant papers: studies describing transformation among business process and Web service representations, and studies exploring the semantic aspects for their integration. Journal, conference, and workshop papers have been considered. A paper selected to be included in this review meets one of the following criteria:

1. It describes a transformation between a business process modeling and a Web service representation.
2. It describes a transformation between different business process representations or between Web service representations. The papers obtained by this criterion do not directly address integration of the two domains; however, they address the heterogeneity of representations, and can facilitate the integration between a business process and a Web service representation.
3. It addresses the semantics aspects of the integration process or of the business process modeling.

The papers obtained by the second criterion do not directly address the integration of business process modeling and Web services; however, they are included as they facilitate their integration. For example, if we consider that there is a transformation approach from BPMN to BPEL, thus a transformation approach from EPC to BPMN will enable a two-step transformation from EPC to BPEL. Therefore, the transformation between the two business process representations, EPC to BPMN, facilitates the integration of business process modeling (EPC) and Web services (BPEL).

Transformations to Petri nets have not been included for two reasons: first, Petri nets are generally considered to be a theoretical modeling approach that is not broadly accepted in practice [24], and second, a survey of Petri net transformations has already been conducted by Lohmann *et al.* [24].

Service composition approaches have also not been included because their main focus is service composition and not transformation among representations. Dustdar and Wolfgang [20], and Rao and Su [21] have conducted surveys on Web service composition.

PSL (Process Specification Language) [25,26] facilitates the exchange of process information among systems; Cheng *et al.* [27] evaluated the applicability of PSL for exchanging project scheduling information among different applications, while Schlenoff *et al.* [28] used PSL to exchange manufacturing process information between a process modeling tool and a scheduling application. PSL is not included as a transformation approach because it does not involve transformation between representations. Nevertheless, we consider PSL particularly relevant for integrating business process modeling and Web service representations and include it when reviewing the use of semantics for integration purposes.

Because of the extent of existing work in semantic Web services [17,18] and in an effort to focus on the integration of business process modeling and Web service representations, semantic Web services are not discussed in depth. However, semantics of business process models are included as essential elements when using Web services for business process execution. Additionally, as this study focuses on the transformation aspects between business process modeling and Web services, studies proposing integrated architecture styles such as Papazoglou and Kratz [29], have not been included.

2.2 Identification of papers

An initial group of papers was identified by searching the INSPEC, Compendex, and IEEE Xplore databases. Searches were performed using combinations of the term "transform" and the abbreviations from the lists of the business process technologies in Subsection 3.1 and Web service technologies from Table 1. Thus, created search phrases were of the form "BPMN transform" and "BPEL transform". The two lists were initially formed by including the major technologies in the two domains and were later expanded as additional technologies involved in the transformations were encountered. From the search results, relevant papers were identified by reading the title and continuing with the abstract if the title was not sufficient.

In an attempt to identify additional papers on the transformation between business processes and Web service representations, Google searches were performed using the same strategy of combining the term "transform" and the abbreviations of technologies. However, Google search identified only a very few relevant papers.

Additional papers were identified using the authors' prior experience and from the reference lists of previously identified relevant papers.

In spite of the efforts to conduct a comprehensive search, it is possible that some papers have been missed. The search was completed in May 2012.

2.3 Threats to validity

The main threats to the validity of this study have been identified as:

Language bias: Only papers written in English have been included.

Selection bias: The search for papers focused primarily on finding research papers, including journal, conference, and workshop papers. Consequently, possibly relevant white papers or technical reports may have been excluded.

Limitation to transformation among business process modeling and Web service representations: Clearly, transformations among representations are encountered in other domains; they are of special interest in model-driven architectures. Findings from these domains could be relevant to transformations between business process models and Web service representations. However, this research has focused on an investigation of proposed approaches for transformation between the business process and Web service domains.

Unpublished transformation studies: Applying a known approach or writing a piece of custom software can also be a means of performing a transformation. Consequently, rather than a research study, a transformation may be considered an implementation challenge and therefore may not have been published in a research paper.

3. Business Process Modeling and Web Services

This section introduces the basic concepts, representation models, and languages used in business process modeling and Web service representations. The heterogeneity of representations in both domains is illustrated.

3.1 Business process modeling

Business process modeling is the activity of representing the business processes of an enterprise so that current processes can be understood, analyzed, and improved. It also includes activities such as design, modeling, execution, monitoring, and optimization. Specifically, business process modeling portrays the business view of the enterprise process. Various business process modeling technologies and languages exist, including:

- BPMN [7], a graphical representation for specifying business processes as Business Process Diagrams (BPD).
- EPCs [8], a flowchart-style representation for business processes.
- YAWL [9], a workflow language based on van der Aalst's workflow patterns [30].
- XPDL [31] (XML Process Definition Language), a format to interchange business process definitions.
- FBPML [32] (Fundamental Business Process Modeling Language), a language with graphical representation and formal semantics. It has been created by the merge of PSL [25,26] and IDEF3 [33].
- BPML [6], a formally complete language according to Pi-calculus [5]. It is being replaced by BEL4WS.
- ebBPSS [34] (ebXML Business Process Specification Schema), an XML-based framework for business process specification.
- IDEF3 [33] (Integration Definition), a method for process flow description and representation of object state transitions.

Although this list is not comprehensive, it illustrates the heterogeneity in business process modeling, specifies terms used for identification of relevant papers, and introduces technologies used in the transformations between different business process models and Web service representations.

Even this concise list demonstrates a great diversity among business process technologies and languages. Business process management has been the subject of surveys conducted by van der Aalst [35] and Ko et al. [36]. Different classifications of business process modeling technologies and languages have been proposed [35,36], nevertheless they typically do not establish clear distinction among categories, but allow for a single language or technology to belong to multiple categories. Van der Aalst [35] distinguishes three language categories: *formal languages*, *conceptual languages*, and *execution languages*. *Formal languages* are built upon theoretical formalisms such as Petri nets and process algebras. They provide unambiguous formal semantics for describing business process models and include languages such as BPML and FBPML. Languages from this category can allow the verification of models such as the verification of YAWL models using the WofYAWL [37] tool. *Conceptual languages* do not have the rigorous semantics of the formal languages; they allow for some vagueness and informality in the modeling. Moreover, languages from this category typically provide robust visual notations and consequently enable convenient and intuitive modeling. Examples of languages from this category are EPC and BPMN. In practice users often prefer languages from this category over formal languages due to the ease of use and informality [35]. BPMN is one of the most frequently used business process representations [38]. *Execution languages* such as BPEL are concerned with business process execution. While van der Aalst [35] considers execution languages as a category of business process languages, we consider them as a Web service technology because these languages typically specify how Web services should implement the business process activities. It is important to point out that languages and technologies can exhibit characteristics from different categories: FBPLM provides formal semantics while providing ease of use along with a graphical representation of the conceptual language.

Knowledge-intensive business services (KIBS) impose additional challenges onto business process modeling as they require modeling human processes which are collaborative, innovative and dynamic in nature. Human process models should not restrict human activities rigidly; they should guide humans while allowing for flexible process execution. Lee *et al.* [39] address this issue by modeling human processes declaratively using pre- and post-conditions [39].

Additionally, to ease the specification of conditions, they apply process patterns. Nevertheless, our work focuses on traditional BPM technologies which support dominantly system-to-system interactions.

Diversity among various business process representations is the cause of significant challenges in integrating business process models and Web services. Instead of dealing with a single business process representation, their integration with Web Services must cope with these numerous heterogeneous representations.

3.2 Web services

Although a number of service-oriented architecture styles have been proposed [40], this work focuses on two common approaches of implementing Web services [17]: services involving SOAP style messages which we refer to as SOAP-based services and RESTful services. SOAP-based Web services are typically accompanied by single-service descriptions written in WSDL, while their composition and orchestration is supported by BPEL and UDDI (Universal Description, Discovery, and Integration). However, recently, there has been a trend towards moving to RESTful services [41], which are occasionally considered the *de facto* standard for service design [42]. BPEL and WSDL have been the accepted approaches to describe single and composite SOAP-based services. For RESTful services, however, a number of approaches have been proposed, including Web Application Description Language (WADL) [12], Protocol for Web Description Resources (POWDER) [43], RESTful Interface Definition and Declaration Language (RIDDL) [44], and HTML for RESTful (hRESTS) [45].

Nonetheless, a number of these technologies, including BPEL, WSDL, and WADL, operate at syntactic level. Therefore, although they support interoperability, these technologies often require human intervention for service discovery and composition [46]. To facilitate service interoperability, discovery and composition, semantic technologies have been adopted to incorporate semantics to traditional Web services. Semantic technologies are crucial for achieving the semantic Web vision in which computers "become capable of analyzing all the data on the Web – the content, links, and transactions between people and computers" [47].

Consequently, to help analyze the transformations among several representations, Web service technologies are classified into semantic and non-semantic categories, as illustrated in Table 1. Furthermore, the two main orientations of Web service styles, SOAP-based services and RESTful services, commonly use technologies which are service style specific. Thus, in Table 1, a further distinction is made between the technologies belonging to the two service styles. Some of the technologies from the same category, such as WSDL and BPEL, are quite dissimilar because their main objectives are different: WSDL describes a single service, while BPEL represents composite executable processes. However, these two technologies are placed into the same category because they are both part of the same SOAP-based service style. In addition, several semantic Web technologies, such as Web Ontology Language for Services (OWL-S), Web Service Modeling Ontology (WSMO), and DARPA Agent Markup Language for Services (DAML-S), are not restricted to a specific Web service style; in Table 1, they are therefore included directly in a semantic technologies category.

Table 1. Web service technologies

		TECHNOLOGY		DESCRIPTION
WEB SERVICES	Non-semantic technologies	SOAP-based services	BPEL4WS [13]	Language for describing process composition. Replaced by WS-BPEL.
WEB SERVICES	Non-semantic technologies	SOAP-based services	WS-BPEL [14]	Language for describing process composition.
WEB SERVICES	Non-semantic technologies	SOAP-based services	WSDL [15]	Language for single Web service description.
WEB SERVICES	Non-semantic technologies	RESTful services	REST [10,11]	Web service architecture style.
WEB SERVICES	Non-semantic technologies	RESTful services	WADL [12]	Language for describing HTTP-based service.
WEB SERVICES	Non-semantic technologies	RESTful services	POWDER [43]	Protocol for describing a group of resources.
WEB SERVICES	Non-semantic technologies	RESTful services	RIDDL [44]	Language for Web service description.
WEB SERVICES	Non-semantic technologies	RESTful services	hRESTS [45]	HTML microformat for describing RESTful Web Services.
WEB SERVICES	Semantic Technologies	DAML-S [52]		Language for describing Web services and related ontologies. Replaced by OWL-S.
WEB SERVICES	Semantic Technologies	OWL-S [53]		Language for describing semantic Web services.
WEB SERVICES	Semantic Technologies	WSMO [54-56]		Framework and language for describing semantic Web services.
WEB SERVICES	Semantic Technologies	WSML [57]		Language for the specification of ontologies and Web services.
WEB SERVICES	Semantic Technologies	SWSF [58]		Framework for describing semantic Web services.
WEB SERVICES	Semantic Technologies	SOAP-based services	SAWSDL [48]	Semantic annotations for WSDL.
WEB SERVICES	Semantic Technologies	SOAP-based services	WSDL-S [49]	Language for describing semantic Web services.
WEB SERVICES	Semantic Technologies	RES	SA-REST [59]	Format for adding annotations to REST API descriptions in HTML or

				XHTML.
			POWDER-S [60]	Semantic POWDER.
			EXPRESS [61]	Approach for expressing RESTful semantic services using domain ontologies.

Similar to the list of business process technologies in Subsection 3.1, Table 1 is not a comprehensive list of technologies, but rather illustrates the heterogeneity of the Web service technologies, specifies terms used for identification of relevant papers, and introduces technologies used in surveyed transformations between business process models and Web services representations. The following paragraphs introduce Web service technologies.

Web Services Description Language (WSDL) [15] is an XML-style language for describing a single service interface. The WSDL description provides information on how the service can be invoked, specifies the location of the service, parameters it expects, and parameters it returns.

Business Process Execution Language (BPEL) [13,14] is an XML-style language for describing composite executable services. The BPEL description specifies how the operations, typically described in WSDL, are composed to form a process in order to achieve a specific business goal. BPEL is concerned with the control flow and the exchange of messages among operations comprising the process. BPEL4WS [13] and WS-BPEL [14] are the recent BPEL variants.

Semantic Annotations for WSDL (SAWSDL) [48] and Web Service Semantics (WSDL-S) [49] define mechanisms for adding semantic annotations to WSDL components. Specifically, they provide a means of referencing semantic models from within WSDL.

Representational State Transfer (REST) is an architecture style governed by a set of constraints applied to the architecture components. The central concept is the notion of resources, each of which is assigned a global identifier such as URI in HTTP. The network components communicate over standardized interface such as HTTP, by exchanging representations of the resources. RESTful services conform to the principles of the REST architecture style.

WADL [12], POWDER [43], RIDDL [44], and hRESTS [45] are approaches for describing RESTful services. Furthermore, BPEL for composition and orchestration of SOAP-based services has been extended to support RESTful service composition [50,51]. The HTTP binding introduced in WSDL 2.0 can be used to wrap RESTful Web services and hide RESTful services inside the WSDL specifications. Consequently, such WSDL wrapped RESTful services can be composed using BPEL. To be able to invoke RESTful Web services directly from BPEL, Pautasso [50,51] extended BPEL by adding the four activities corresponding to the HTTP methods: <get>, <post>, <put>, and <delete>. Thus, instead of relying on WSDL, the new constructs directly map to the HTTP methods.

RESTful specific semantic technologies include Semantic annotations for REST (SA-REST) [59], Semantic POWDER (POWDER-S) [60], and EXPRESS [61]. SA-REST provides a means of adding annotations to REST API descriptions in HTML or XHTML. POWDER-S provides formal semantics for POWDER. Specifically, POWDER-S aims to make POWDER data available for Semantic Web tools by representing POWDER data as an OWL ontology. The EXPRESS approach specifies RESTful semantic services using domain ontologies. Subsequently, RESTful interfaces are created from ontology classes, instances and relations.

Generic semantic technologies include OWL-S [53], DAML-S [52], WSMO [54-56], WSML [57], and SWSF [58]. Semantic Web Services Framework (SWSF) aims to provide support for achieving semantic Web vision by provisioning for richer semantics, expressive representations of various aspects of services, and automated service composition.

Because information semantics form a crucial part of the integration of business process modeling and Web services, this paper introduces in more detail generic semantic Web service technologies frequently involved in transformations: OWL-S, DAML-S, and WSMO with its representation language WSML.

OWL-S and DAML-S are ontologies for describing semantic Web services; however, DAML-S has been superseded by OWL-S. The structure of OWL-S is composed of three parts [53]:

a) *Service profile* describes what the service does so that consumers can determine whether the service meets their needs. It includes elements such as service name, a description of service function, limitations on applicability, quality of service, and the requirements that the service consumer must meet. Its objective is service advertising and discovering.
b) *Process model* describes how to use the service by specifying the interactions with the service in a semantic way. It includes elements such as inputs, outputs, service's operations, pre- and post-conditions. The service consumers use the process model to

verify that the service meets their needs, to compose it with other services, to invoke the service, and to monitor its execution.

c) *Grounding* provides information about how to invoke the service including the communication protocol, message formats and port numbers.

Another semantic Web technology, Web Service Modeling Ontology (**WSMO**), is a conceptual framework and formal language for describing semantic Web services [54,56]. Specifically, the formal language used by WSMO is Web Service Modeling Language (WSML). WSMO framework consists of four main elements [55]:

a) *Ontologies* provide the terminology used by the other WSMO elements.
b) *Web services* provide access to services through an explanation of the services' capabilities, interfaces, and operations.
c) *Goals* describe users' desires.
d) *Mediators* manage interoperability problems among the WSMO elements.

Both OWL-S and WSMO aim to describe semantic Web services and to automate service discovery, composition, interoperation, and invocation; however, these technologies use different approaches to achieve this goal. The main differences between OWL-S and WSMO are highlighted here, while Lara *et al.* [62] have discussed in detail the conceptual differences between the two. OWL-S does not distinguish between the elements that describe what the user wants and what the service provides; both are represented by the service profile. In WSMO, the goal describes the user's desires while the Web service capabilities specify what the service offers. WSMO recommends the use of specific vocabularies for describing non-functional properties; whereas OWL-S does not impose such restriction. Moreover, OWL-S requires a rule language such as Semantic Web Rule Language (SWRL) for describing logical expressions, whereas the WSMO specification language WSML (Web Service Modeling Language) intrinsically combines conceptual modeling with rules. Overall, OWL-S is considered more mature, especially in respect to choreography and grounding specifications, while WSMO attempts to provide a more complete model [63].

Guo [64] argues that process-oriented approaches such as BPEL and OWL-S do not solve issues such as security, trust, and flexible interaction handling. Consequently, he proposed a service-based multi-agent platform. However, agent-based services are outside the scope of this work.

4. Transformation among Business Process Representations and Web Service Technologies

The diversity of business process modeling and Web service technologies, as illustrated in Section 3, represents an obstacle to automating business process execution through the use of Web services. Each technology demonstrates strengths in some aspects of business process modeling or Web services and displays weaknesses in others. For example, BPEL is effective as an execution language; however, it does not have the semantic abilities of OWL-S or the graphical representation capabilities of FBPML. Therefore, a variety of technologies remains in use, and attempts are continuously made to transform from one to the other.

4.1 Transformation studies

Relevant literature involving attempts to transform between different process representations was obtained using the search strategy described in Subsection 2.2. Table 2 presents an overview of the papers identified, including the direction of the transformation they address. Although this research is primarily focused on the transformation between business process and Web service representations, it also includes transformations between languages belonging to the same category of business process specifications or Web service representations. In particular, transformations between two representations in the same category are significant because they address the issue of heterogeneity of representations and consequently, they facilitate integration of business process modeling and Web services as described in Subsection 2.1.

In Table 2, in addition to distinguishing between Web services and business process technologies, Web services have been further divided into two categories: semantic and non-semantic Web services.

The transformation of ARIS markup Language (AML) to the EPC Markup Language (EPML) [65] has not been included in this survey as AML is a proprietary interchange format of the ARIS toolset, thus, this transformation is tool-specific.

Transformations to non-semantic Web services. The transformation from business process specifications to non-semantic Web services, specifically involving different BPEL variants, has attracted significantly more research attention than transformations among other categories. Motivations for such attention include the benefits of this transformation direction in enabling the execution of business process specifications and the popularity of BPEL for the implementation of composite Web services [67,69,88].

The number of proposed technologies for RESTful service is vast; it includes WADL, POWDER, RIDDL, SAWSDL, SA-REST, and others, as illustrated in Table 1. However, as shown in Table 2, only two transformation studies involving RESTful services were found: one by Upadhyaya *et al.* [77], presenting an approach for migrating SOAP-based services to RESTful services, and the other by Peng *et al.* [78] describing a framework which wraps RESTful services inside SOAP-based services, enabling the creation of a BPEL specification by combining the two service styles. Moreover, no studies involving a transformation between business processes representations and RESTful services were found. A possible explanation for this is the absence of a broadly accepted way for describing RESTful services. In recent years, a number of approaches have been proposed to describe RESTful services including WADL, SA-REST, hRESTS, POWDER, and RIDDL; however a dominant approach has not emerged [17].

Transformations to semantic Web services. Transformations to semantic Web services representations OWL-S, DAML-S, and WSMO from a variety of process representations have also attracted substantial attention. However, this category is dominated by transformations from non-semantic Web services representations BPEL and WSDL: Shen *et al.* [69,72], Aslam *et al.* [70], Wang *et al.* [71], and Paolucci *et al.* [76].

Transformations from business process representations to semantic Web services were considered only in the studies by Nadarajan and Chen-Burger [2] and Guo *et al.* [84]. This could be an indication of a trend towards moving from non-semantic to semantic Web services; however, this information by itself is not sufficient to draw such a conclusion.

Transformations to business process representations. Transformations from non-semantic Web services, specifically different BPEL variants, to different business process representations have been addressed in studies by Brogi and Popescu [73], Mendling and Ziemann [75], Norton *et al.* [3], and Weidlich *et al.* [74]. However, no studies were found involving a transformation from semantic Web services to a business process specification.

Transformations between models of the same category. This category deals with representation heterogeneity within domains and includes transformations between two business process models and between two Web service representations. This category is indicated by shaded fields in Table 2. Scicluna *et al.* [66] and Le *et al.* [68] have addressed the representation heterogeneity of semantic Web services, Upadhyaya *et al.* [77] and Peng *et al.* [78] that of non-semantic services, and Ye *et al.* [82], Decker *et al.* [83], and Vanderhaeghen *et al.* [87] that of business process representations.

Table 2. Transformations between business process representations and Web service technologies

			TRANSFORMATION TO									
			Web Services						Business Process Representations			
			Semantic Web Services			Non-semantic Web Services						
			OWL-S	DAML-S	WSMO	BPEL (sBPEL, BPEL4WS)	RESTful	SOAP-based service	BPMO	YAWL	BPMN	EPC
TRANSFORMATION FROM: Web Services	Semantic Web Services	OWL-S			Scicluna *et al.* [66]	Bordbar *et al.* [67]						
		WSMO	Le *et al.* [68]									
	Non-semantic Web Services	BPEL (sBPEL, BPEL4WS)	Shen *et al.* [69], Aslam *et al.* [70], Wang *et al.* [71]	Shen *et al.* [72]					Norton *et al.* [3]	Brogi and Popescu [73]	Weidlich *et al.* [74]	Mendling and Ziemann [75]
		WSDL		Paolucci *et al.* [76]			Upadhyaya *et al.* [77]					
		RESTful						Peng *et al.* [78]				
TRANSFORMATION FROM: Business Process Representations		BPMO				Norton *et al.* [3], Cabral and Domingue [79]						
		BPMN				Ouyang *et al.* [1,38], García-Bañuelos [80], BPMN [7,81]				Ye *et al.* [82], Decker *et al.* [83]		
		FBPML	Nadarajan and Chen-Burger [2], Guo *et al.* [84]									
		EPC				Ziemann and Mendling [85], Meertens *et al.* [86]					Vanderhaeghen *et al.* [87]	

To allow the examination of the transformation directions in greater depth, Table 3 presents the information contained in Table 2 from a different perspective; specifically it correlates the studies addressing transformations in opposite directions. Note that only one publication, Norton *et al.* [3], includes a bidirectional transformation, specifically from BPMO (Business Process Modeling Ontology) to BPEL and from BPEL to BPMO. Ziemann and Mendling have investigated an EPC-to-BPEL transformation as well as its reverse transformation; however, they presented them in two different publications [75, 85].

In the case of several pairs of representations, such as WSMO and OWL-S or BPEL and OWL-S, different authors have studied different directions of transformation. For example, Le *et al.* [68] considered a transformation from WSMO to OWL-S, while Scicluna *et al.* [66] investigated a transformation in the opposite direction, from OWL-S to WSMO. Although the opposite transformations may exhibit some similarities, such as mapping similarities between the work of Le *et al.* [68] and that of Scicluna *et al.* [66], the findings for one transformation are not always considered when studying the opposite direction.

Moreover, several transformations, such as BPMN to YAWL and BPEL to YAWL, do not have a corresponding opposite transformation. This indicates that even though a transformation in one direction was attempted, the strategy for identification of papers described in Subsection 2.2 did not locate studies addressing their corresponding opposite transformation.

A complete transformation from one business process or Web service representation to another and back is rarely possible due to the great differences in their objectives and capabilities. However, we believe that the findings from one direction of transformation should be used when observing the reverse direction. Considering both transformation directions at the same time would lead to better reuse of research findings and would highlight the variations and similarities of different representations.

4.2 Transformation approaches

Transformation studies from Table 2 were further examined with regard to the approach used; the findings are presented in Table 4. With regard to transformation approaches, it is important to note that each of the proposed approaches except for that of Vanderhaeghen *et al.* [87] addresses a specific pair of technologies. Addressing all possible transformations by considering independently each potential representation pair would be impractical because the number of possible pairs is extensive; only a small fraction is presented in Table 2. Vanderhaeghen *et al.* [87] proposed a generic transformation process independent of the source and target representations; however, their approach requires XML representations of the source and target models.

Table 3. Transformation directions

TRANSFORMATION IN ONE DIRECTION		**TRANSFORMATION IN THE OPPOSITE DIRECTION**	
Transformation direction	Study	Opposite transformation direction	Study
WSMO to OWL-S	Le *et al.* [68]	OWL-S to WSMO	Scicluna *et al.* [66]
BPEL to OWL-S	Shen *et al.* [69] Aslam *et al.* [70] Wang *et al.* [71]	OWL-S to BPEL	Bordbar *et al.* [67]
BPMO to BPEL	Norton *et al.* [3] Cabral and Domingue [79]	BPEL to BPMO	Norton *et al.* [3]
BPMN to BPEL	Ouyang *et al.* [1,38] García-Bañuelos [80] BPMN 1.2 [7]	BPEL to BPMN	Weidlich *et al.* [74]
EPC to BPEL	Ziemann and Mendling [85] Meertens *et al.* [86]	BPEL to EPC	Mendling and Ziemann [75]
WSDL to RESTful	Upadhyaya *et al.* [77]	RESTful to SOAP-based (not complete opposite)	Peng *et al.* [78]
BPEL to DAML-S	Shen *et al.* [72]		
WSDL to DAML-S	Paolucci *et al.* [76]		
FBPML to OWL-S	Nadarajan and Chen-Burger [2] Guo *et al.* [84]		
BPEL to YAWL	Brogi and Popescu [73]		
BPMN to YAWL	Ye *et al.* [82], Decker *et al.* [83]		
EPC to BPMN	Vanderhaeghen *et al.* [87]		

Table 4. Transformation approaches

Transformation approaches			
Mappings and transformation described primarily in freeform text		Rules	Other
Mappings and transformation described in text form	Text with mapping lists and/or tables		
• García-Bañuelos *et al.* [80] • Shen *et al.* [69] • Shen *et al.* [72] • Brogi and Popescu [73] • Decker *et al.* [83] • Paolucci *et al.* [76] • Ziemann and Mendling [85] • Mendling and Ziemann [75] • Guo *et al.* [84]	• Weidlich *et al.* [74] • Scicluna *et al.* [66] • Wang *et al.* [71] • Le *et al.* [68] • Ye *et al.* [82] • Ouyang *et al.* [1,38] • Aslam *et al.* [70]	• Cabral and Domingue [79] (ATL rules) • Bordbar *et al.* [67] (SiTra framework) • Vanderhaeghen *et al.* [87] (XSLT rules)	• Norton *et al.* [3] (Mappings expressed in Ontologis) • Meertens *et al.* [86] (Focus on transformation feasibility) • Nadarajan and Chen-Burger [2] (Transformation maps between source and target ontologies) • Upadhyaya *et al.* [77] (Dependency graph and clustering) • Peng *et al.* [78] (Wraps RESTful services into SOAP-based service)

4.2.1. Mappings and transformations described primarily in freeform text

The majority of the transformation studies have focused on finding entities in one representation that can be directly correlated with the elements of another model. In particular, a large number of studies have described these mappings and the transformation process in freeform text without any formalization. In Table 4, these studies are identified as "Mappings and transformation described in text form". The next category, "Text with mapping lists and/or tables", uses lists, tables, or both in addition to freeform text to represent the mappings. Even though there is no major difference between these two described categories, being both procedural, they are presented separately because we believe that lists and tables facilitate comprehension.

Table 5 illustrates mappings on an example drawn from the study by Shen *et al.* [72] on transforming BPEL/WSDL to DAML-S. BPEL/WSDL *abstract* processes are mapped to DAML-S *simple* processes and *executable* processes are mapped to *atomic* and *composite* processes. Furthermore, the *activities* making up the building blocks of BPEL are mapped as follows: *primitive* activities are mapped to *atomic* processes, while *structured* activities composed of other primitive or structured activities are mapped to *composite* processes of DAML-S.

In the work of Shen *et al.* [72] only the *process* mappings of Table 5 were presented in a table form; while the *activity* mappings were described textually. Because OWL-S is the successor of DAML-S, Shen *et al.* used the same mapping in their work to transform BPEL to OWL-S [69]. Moreover, Aslam *et al.* [70] used the same mapping, with the exception of abstract processes, which they did not address.

Scicluna *et al.* [66] also focused on a conceptual mapping; the OWL-S components and their properties are mapped to corresponding WSMO constructs. As in OWL-S, a service is described using the three parts of the language: *service profile*, *process model* and *grounding*; the mapping is addressed separately for each of these three parts as depicted in Table 6. The OWL-S *service profile* describes the service functionalities, and therefore, it is mapped to the WSMO *capability*. A *process model* defines the way in which the service works in terms of its methods and their coordination; thus, the *process model* is mapped to the components *capability* and *choreography* of WSMO. However, the OWL-S *grounding* is not mapped to WSMO, as there is no strict corresponding entity in WSMO. The mapping of flow control constructs was not addressed in the work of Scicluna *et al.* [66].

Table 5. Mappings from BPEL/WSDL to OWL-S elements

Source: BPEL or WSDL element		Target: DAML-S element
Process	Abstract process	Simple process
	Executable process	Atomic and composite processes
Activity	Primitive activity	Atomic process
	Structured activity	Composite process

Table 6. Conceptual mapping between OWL-S and WSMO

OWL-S	WSMO
service profile	*capability*
process model	service *capability* and *choreography*
grounding	-

Le *et al.* [68], like Scicluna *et al.* [66], examined the conceptual mapping between WSMO and OWL-S; however, they considered a different direction of transformation, that is, from WSMO to OWL-S. Similar to the work of Scicluna *et al.* [66], the work of Le *et al.* [68] did not include OWL-S grounding. As WSML and OWL are the underlying ontology languages of the WSMO and OWL-S specifications, in the approach of Le *et al.* [68] the transformation from WSML to OWL was performed as part of the WSMO to OWL-S transformation. WSML and OWL consist of similar entities but with distinct terms as illustrated in Table 7. The WSML to OWL transformation can be considered as a distinct transformation and can be used independently in problems such as ontology mapping and merging.

Establishing mappings can be beneficial in determining the limitations of transformations between specific pairs of representations. For instance, Table 8 shows partially the mapping BPEL-to-BPMN described by Weidlich *et al.* [74]. The third column indicates whether a concept is mapped directly (+), the concept is missing in BPMN (-), or the mapping is partial (°) to BPMN. Weidlich *et al.* [74] identified the three activities that cannot be mapped at all: *event handlers*, *termination handlers* and *validate* activities. Table 8 includes only one unmapped activity; *validate*, as the table illustrates the mapping for basic activities only. Several constructs, can only be mapped partially. Weidlich *et al.* [74] considered a mapping is partial if it was incomplete or imprecise or if an approximation was performed in the transformation. An example is the BPEL concepts of *reply* which is associated with the received message by the concept of *message exchange*. Specifically, the BPEL concept of *message exchange* correlates messages belonging to the same conversation. The counterpart is absent from BPMN; however, it can be modeled using the concept of properties [74] which can be assigned to messages. Specifically, a dedicated property assigned to a message can be used to correlate messages belonging to the same conversation.

The majority of the studies, described in this Section so far, focused mainly on conceptual mappings between pairs of representations. In addition to the already mentioned studies of Shen *et al.* [72], Aslam *et al.* [70], Scicluna *et al.* [66], Le *et al.* [68], and Weidlich *et al.* [74], a number of other studies also focused on conceptual mappings including Ziemann and Mendling [85], Mendling and Ziemann [75], and Paolucci *et al.* [76]. The works of Ouyang *et al.* [1,38], García-Bañuelos *et al.* [80], and Brogi and Popescu [73], however, considered the transformation process in addition to the mappings.

Ouyang *et al.* [38] proposed a formalized approach for the transformation process from BPMN to BPEL. As groundwork for their transformation process, Ouyang *et al.* divide the BPMN's Business Process Diagrams (BPD) into segments allowing for an iterative approach where the BPMN components are incrementally transformed into BPEL entities. The transformation consists of two steps: the first step, the activity-based transformation includes mapping the well-structured BPD components such as sequence, flow, pick, while, repeat, and repeat+while to the corresponding BPEL structured activities including sequence, flow, switch, pick, and while. The second step involves translating the remaining BPD components using event-action rule-based transformations. Although event-action rule-based transformations can be used for all components, Ouyang *et al.* [38] use these transformations only for components that are not well-structured, as these transformations result in BPEL code that is difficult to read.

García-Bañuelos [80] work focused on the identification of patterns in BPMN graphs that can be transformed straightforwardly to BPEL specifications. Like Ouyang *et al.* [38], his approach first partitions a BPMN graph into segments, in this case into single-entry single-exit (SESE) regions. Next, control flow analysis is performed on the SESE regions and patterns are identified for each SESE region. Subsequently, the SESE regions are transformed into sub-processes that are annotated with the identified patterns. The final step of his approach transforms the annotated sub-processes into BPEL by using translation rules.

Table 7: WSML to OWL mapping

WSML	OWL
concept	*class*
relation	*property*
relation attributes such as *transitive* or *inverse property*	described using *axioms*
individuals	*instances*

Table 8. Mapping from BPEL to BPMN

	BPEL	BPMN	Mapping
Basic Activities	*invoke*	sending/receiving task, message event	°
	receive(*createInstance='no'*)	receiving task, message event	+
	reply	sending task, message event	°
	validate	-	-
	assign	assignment	°
	wait	timer intermediate event	+
	exit	termination end event	+
	throw, rethrow	error end event	°
	compensate, compensateScope	compensation events	°

Brogi and Popescu [73] approach is based on defining YAWL patterns for the BPEL processes and their activities. The three main patterns, namely, *basic*, *structured* and *process* patterns correspond to the BPEL *basic* activity, *structured* activity and *process* respectively. Each BPEL activity or process is instantiated from its corresponding pattern. The process of instantiating a pattern, described by Brogi and Popescu [73], involves adjusting inputs and outputs for each activity as well as creating connections between patterns.

4.2.2. Rule-based approaches

The next transformation approach category from Table 4, "Rules", includes approaches which use rules to represent the transformation: Cabral and Domingue [79] used the Atlas Transformation Language (ATL) rules, Bordbar *et al.* [67] applied the Simple Transformer (SiTra) framework, and Vanderhaeghen *et al.* [87] used the eXtendable Stylesheet Language Transformation (XSLT). The work of Cabral and Domingue [79] and Bordbar *et al.* [67], considered a particular pair of representations: Cabral and Domingue [79] considered BPMO to BPEL, while Bordbar *et al.* [67] considered OWL-S to BPEL transformation. Unlike them, Vanderhaeghen *et al.* [87] proposed a generic procedure for transformations between different business process representations.

Cabral and Domingue [79] considered a transformation approach from BPMO to BPEL. The transformation source entails instances of BPMO in a WSML file representing specific business processes, while the target is the WSML file containing the corresponding instances in sBPEL. The proposed approach uses Atlas Transformation Language (ATL) for the representation of the transformation rules which in fact express mappings between the elements of the source (BPMO) and the target system (BPEL).

The main focus of the work from **Bordbar *et al.*** [67] entailed exploring the capabilities of the Simple Transformer (SiTra) framework, while the transformations from OWL-S to BPEL were examined in a case study. The first step in the transformation performed in the SiTra framework involves the creation of meta-models for the source and target models, which, in their case study are the OWL-S and BPEL meta-models. SiTra transformation rules describe how entities of the source meta-model are transformed to the elements of the target model. The transformation process converts instances of the source model to instances of the target model by matching rules with applicable source objects, executing the rules and creating objects in the target model. The rule specification requires establishing mappings between the source and target elements, which, in their use case, constitute mappings between the OWL-S and BPEL entities.

The generic approach for transformations among business process representations described by **Vanderhaeghen *et al.*** [87] requires an XML description for each representation involved in the transformation. In their example case of a transformation from EPC to BPMN [87], the XML representations were EPML and BPML. The first phase established the relation between the source and target representations and involved the following steps:

1. establishing meta-models for each representation,
2. mapping the meta-models in which elements of the source meta-model were mapped to components of the target model.

The second transformation stage entailed creating a source model XML representation, transforming the source XML model to the target XML, and finally transforming the target XML to the target model.

The main drawback of this approach is the need for XML representations [87] of the source and target models, which results in a multi-step transformation: from source model to source XML, source XML to target XML, and target XML to target representation. Although Vanderhaeghen *et al.* [87] facilitated the second transformation phase, from source XML to target XML, by using XSLT rules, the other two transformations remain a challenge as they still need to be addressed individually for each pair of representations. Vanderhaeghen *et al.* demonstrated a transformation from EPC to its XML representation EPML and from BPMN's XML representation BPML to BPMN; however, they do not indicate how this step would be performed in the case of dealing with different representations.

4.2.3. Other approaches

The remaining studies used highly diverse approaches and therefore are placed in the "Other" category in Table 4.

Norton *et al.* [3] considered a bidirectional transformation and proposed ontology-based associations between different representations in which the source and target models, BPMO and BPEL, were represented as ontologies. The transformations were based on ontologies: BPMO2sBPEL represents the transformation from BPMO to BPEL, and BPEL2BPMO denotes the transformation from BPEL to BPMO. Both ontologies imported representations of BPMO and BPEL models into Web Service Modeling Language (WSML). The transformation rules could therefore be represented using WSML-Flight axioms that enable the deduction of target instances from source instances.

Meertens *et al.* [86] proposed a generic framework for evaluating the feasibility of transformations between different modeling languages and evaluated this framework using the EPC-to-BPEL transformation. The framework consisted of two parts:

- The first part, an ontological analysis, consisted of evaluating the two languages involved in the transformation against the Bunge-Wand-Weber (BWW) [89] model and subsequently comparing the two languages to each other. This step exposed the limitations and challenges of the transformation, such as source concepts that are not supported in the target language as well as redundancies and overloads.
- The second step, workflow pattern support, consisted of evaluating each language against the twenty Workflow Control Patterns (WFCP) from van der Aalst [30] and then performing comparisons between the two languages. This step exposed patterns that might cause challenges in the transformation process.

Unlike other transformation studies, Meertens *et al.* focused on transformation feasibility rather than on the mapping or the transformation process itself.

Nadarajan and Chen-Burger [2] provided a framework for FBPML to OWL-S transformation consisting of data model and process model transformations. FBPML and OWL-S data and process models were represented using ontologies. Additionally, mapping principles were used to transform from one ontology model to another:

- Data model transformation: The FBPML data model was described using FBPML Data Language (FBPML DL), while the OWL-S data model was delineated in Web Ontology Language (OWL). The data model transformation was performed according to mappings between the ontology representation of the FBPML data model and the OWL-S data model.
- Process model transformation: The FBPML Process Language (FBPML PL) was used to describe the FBPML process model, while OWL-S contained classes delineating the process model. The process model transformation was carried out according to mappings between the ontology representation of the FBPML process model and the OWL-S process models.

Upadhyaya *et al.* [77] **and Peng *et al.*** [78], in contrast to most of other transformation approaches discussed in the paper, did not rely on entity-level mappings. The fundamental differences between the two representations they considered, SOAP-based to RESTful services, did not permit such mappings. To migrate SOAP-based to RESTful services, Upadhyaya *et al.* [77] built a dependency graph from a WSDL document, clustered similar operations, analyzed each cluster to identify resources and HTTP methods, and subsequently created a RESTful wrapper for SOAP-based services. The opposite direction of the transformation, RESTful to SOAP-based services, was considered by Peng *et al.* [78]. In particular, they used WADL description of RESTful service to wrap RESTful service into SOAP-based services.

4.3 The role of ontologies in transformations

The aim of this subsection is to examine the role of ontologies in the transformation approaches described in Table 2. Ultimately, this analysis will show that ontologies can be used to describe source and target models as well as the transformation process itself.

The semantic Web service technologies that occur in the observed transformations, specifically OWL-S, DAML-S, and WSMO (Table 2), are ontologies for describing Web services. In addition, another technology involved in the observed transformations is also an ontology: BPMO (Business Process Modeling Ontology) is an ontology for describing business process models. Consequently, as illustrated in Table 9, in a large number of transformations, the source or the target is either an ontology or an ontology language: BPMO, OWL-S, DAML-S, or WSMO.

Moreover, some approaches have used ontologies in the transformation process. For instance, Norton *et al.* [3] represented the source and target representations, BPMO and BPEL, as ontologies, while the transformations were also performed by means of transformation ontologies: BPMO2sBPEL and BPEL2BPMO. BPMO2sBPEL transformation ontology captures the mapping from BPMO to sBPEL whereas BPEL2BPMO captures the opposite direction, from BPEL to BPMO. The BPEL ontology, which is the conceptualization of the BPEL specification, allows Norton *et al.* [3] to create ontology-based representation of the BPEL processes. Both, BPMO and BPEL ontologies were represented in WSML, thus the transformation rules can be expressed as WSML-Flight axioms. In the case of the transformation from BPEL to BPMO, Norton *et al.* [3] first create the ontology-based representation of the BPEL process, and then transform it to the BPMO representation using the transformation ontologies.

Table 9: The role of ontology in transformations

Source or target models are ontologies	Transformation	Study	Ontology involved in transformation
YES	BPMO to BPEL	Cabral and Domingue [79]	
	Between BPMO and BPEL	Norton *et al.* [3]	Source and target models expressed as ontologies. Transformations performed by means of ontologies: BPMO2sBPEL and BPEL2BPMO
	BPEL/WSDL to DAML-S	Shen *et al.* [72]	Source and target meta-models can be seen as ontologies.
	OWL-S to BPEL	Bordbar *et al.* [67]	Source and target meta-models can be seen as ontologies.
	BPEL to OWL-S	Shen *et al.* [69]	
		Aslam *et al.* [70]	
		Wang *et al.* [71]	
	FBPML to OWL-S	Nadarajan and Chen-Burger [2]	Source and target models expressed as ontologies
		Guo *et al.* [84]	
	WSDL to DAML-S	Paolucci *et al.* [76]	
	WSMO to OWL-S	Le *et al.* [68]	
	Between OWL-S and WSMO	Scicluna *et al.* [66]	
NO	BPMN to BPEL	Ouyang *et al.* [38]	
		García-Bañuelos *et al.* [80]	
	BPEL to BPMN	Weidlich *et al.* [74]	
	BPEL to YAWL	Brogi *et al.* [73]	
	BPMN to YAWL	Ye *et al.* [82]	
		Decker [83]	
	EPC to BPEL	Ziemann and Mendling [85]	
		Meertens *et al.* [86]	
	BPEL to EPC	Mendling and Ziemann [75]	
	EPC to BPMN	Vanderhaeghen *et al.* [87]	Source and target meta-models can be seen as ontologies.
	WSDL to RESTful services	Upadhyaya *et al.* [77]	
	RESTful to SOAP-based	Peng *et al.* [78]	

The Nadarajan and Chen-Burger [2] introduced a framework involving transformations of the data and process models in which the source and target models, FBPML and OWL-S, were represented using ontologies. Vanderhaeghen *et al.* [87], Shen *et al.* [72], and Bordbar *et al.* [67] used source and target meta-models, which can be perceived as ontologies.

4.4 Transformation benefits

All transformation studies included in Table 2 contribute to the integration of business process modeling and Web services. However, the specific benefits of different transformations differ as shown in Table 10. For the purpose of analyzing the benefits of different transformations, these transformations were grouped into four categories. The first one includes transformations without any change in process view perspective, while the classification criterion for the remaining three categories is the transformation target: to semantic Web services, to non-semantic Web services, and to business process representations. Nevertheless, within each of these four categories, the benefits vary; thus Table 10 further subdivides these transformations into subcategories in order to group them with common benefits.

The studies conducted by Ye *et al.* [82], Decker *et al.* [83], and Vanderhaeghen *et al.* [87] are all from the same category in Table 2. The benefits of such transformations between business process specifications consist of managing issues of representation heterogeneity. Moreover, when participants within collaborative networks use different modeling methods, the transformations between business process specifications facilitate the exchange of business process models among participants.

Similarly, the benefits of studies involving a transformation to semantic Web services [2,69-72,76,84] lie in providing well-defined semantics and in facilitating automated Web service discovery and composition.

In Table 10, transformations to YAWL are divided into its own category to emphasize the specific benefit of this transformation: transformation to YAWL enables model verification using YAWL verification tools, such as WofYAWL [37]. This category includes the works of Brogi and Popescu [73], Ye *et al.* [82], and Decker *et al.* [83]. Moreover, transformation to YAWL also belongs to the subcategory made up of transformations from BPEL to business process representations and therefore also offers the benefits specified for this subcategory: enabling visualization of an existing BPEL model, facilitating transformations from executable to business process models and vice versa, and assisting alignment of representations through mappings.

Table 10. Transformation benefits

Transformation category	Transformation subcategory	Benefits
Transformation without change in process view perspective	• Transformation among business process specifications	• Managing the representation heterogeneity challenge • Exchanging business process models among participants within collaborative networks
	• Transformation among Web service representations	• Managing the representation heterogeneity challenge • Facilitating Web service discovery and composition
Transformation to semantic Web services	• Transformation from business process specification to semantic Web service representation • Transformation from non-semantic to semantic Web service representation	• Providing well-defined semantics • Facilitating automated Web service discovery and composition
Transformation to non-semantic Web services	• Transformation from business process specification to non-semantic Web service representation	• Enabling execution of business process representations
	• Transformation from OWL-S to non-semantic Web service representation	• Expressing semantic Web services in widely accepted BPEL [67,69,88], which Bordbar *et al.* [67] claim to provide better tools and support for execution
Transformation to business process representation	• Transformation from BPEL to business process specification (including transformation to YAWL)	• Enabling visualization of an existing BPEL model, thus facilitating communication of existing BPEL processes to business analysts for approval or re-engineering • Facilitating transformations from executable to business process models and vice versa • Assisting alignment of representations through mappings
	• Transformation to YAWL specifically	• Enabling model verification using YAWL verification tools such as WofYAWL

4.5 Transformation trends

This section examines the trends and directions of the transformations that bridge the gap between business process representations and semantic Web services. Two views of the process are involved: the business view represented by business process models and the executable view related to Web services. Therefore, transformations that move towards a business process representation can be distinguished from those moving towards Web services.

The changes in the process view perspective for different transformations are presented in Figure 1. The *degree of change in a process view perspective* refers to the difference in the process view between the source and its target representations. The degree of change is not a measurable attribute, but an approximate and relative value. It is shown in Figure 1 without numerical values on the vertical axis, making it possible to observe transformation trends. If the source is a business process model and the corresponding target is a semantic Web service technology, the degree of change in the process view perspective is large. If both source and target belong to the same category, meaning that both are business process models or both are Web service technologies, there is no change in the process view perspective.

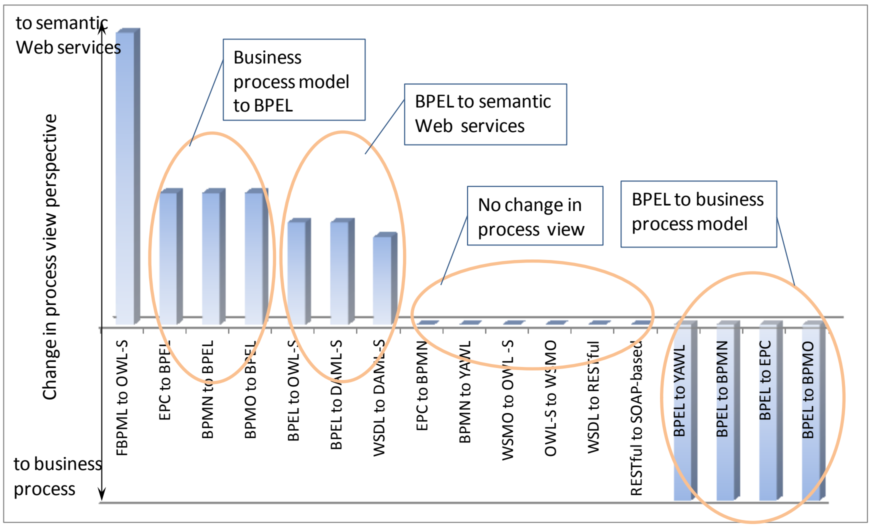

Fig. 1 Transformation trends

The largest change in process view perspective occurs in the FBPML to OWL-S transformation, which involves two representations on opposite sides of the spectrum: a business process model (FBPML) and a semantic Web service (OWL-S).

A relatively major change in process view perspective also occurs in four transformations that move towards business process representations. Each one involves BPEL as a source model, and the target models consist of various business process representations: YAWL, BPMN, EPC, and BPMO. The degree of change is considered smaller in these than in the FBPML to OWL-S transformation because BPEL is a non-semantic Web service technology.

Transformations from business process representations, EPC, BPMN, and BPMO, to BPEL involve a move towards semantic Web services because they are changing from a business process representation to a Web service representation. However, the BPEL target is a non-semantic technology, and therefore the degree of change is smaller than in those pairs involving business processes and semantic Web services, specifically FBPML to OWL-S.

In addition, transformations from BPEL/WSDL to semantic Web technologies, OWL-S and DAML-S, are considered transformations towards semantic Web services. In this case, both source and target are Web service technologies; however, the source is non-semantic, while the target is semantic. Consequently, the transformation involves a move towards semantic Web technology, but the degree of change is relatively small because both are Web service technologies.

The OWL-S to BPEL transformation is not included in the diagram because its orientation is away from semantic Web services, and therefore it cannot be shown in the upper part of the diagram. At the same time, it cannot be included below the horizontal axis because it does not involve a business process representation.

Another large group includes transformations that do not change the process view perspective. Rather, they operate between different models in the same category: between business process representations (EPC to BPMN and BPMN to YAWL), between semantic Web services (WSMO to OWL-S and OWL-S to WSMO), or between non-semantic Web services (WSDL to RESTful and RESTful to SOAP-based service).

Fig. 1 shows that pairs of representations involving a transformation towards semantic Web services have attracted more research attention than those involving a transformation toward a business process representation. This imbalance has been primarily driven by the need to make business processes executable with minimal human involvement. However, only one of the pairs shown includes a business process and a semantic Web service model, FBPML to OWL-S. All other pairs from this category involve a non-semantic Web service technology, BPEL/WSDL.

Moreover, transformations among representations in the same category have also attracted considerable research attention. In large part, these transformations have been motivated by the need to overcome heterogeneity of representations. For Web services, this includes enabling a service described in one language to be matched to a service request represented in another language; while for business process representations the transformations facilitate model exchange and integration.

Transformations oriented towards business process representations involve BPEL as a source for various target business process models. This direction is primarily motivated by the need to represent existing BPEL models in a graphical form.

4.6 Verification techniques

Once the transformations between representations are completed, verification can confirm that the resulting representation exhibits the desirable characteristics. A thorough review of verification techniques is outside the scope of this work; however, this subsection introduces techniques for model verification as they are essential for ensuring the correct behavior of the transformed processes.

This study observes transformations in two directions: from business process representations to Web services and from Web services to business process representations. Consequently, verification of the two resulting categories should be observed. Since Web services are responsible for the execution of the business processes, we mainly discuss the verification of Web services.

The verification of Web services involves composite services with the objective of ensuring that the execution produces the desired behavior. Due to the popularity of BPEL for the implementation of composite Web services [67,69,88], a number of verification approaches addresses BPEL specifically [90,91]. Hull and Su [92] indicate that an important step in analyzing BPEL services is the transformation into formalisms that are better suited for analysis such as finite state machines, extended Mealy machines and process algebra. Thus, in model verification, the transformation is often a step in the verification process while here is the main focus of our study.

Research conducted by Lomuscio *et al.* [90] and Yeung [91] exemplify the verification process of BPEL models. Lomuscio *et al.* [90] study the verification of the behavior of agent-based composite services which is regulated by contracts. All the possible agent behaviors, as well as the correct behavior according to contracts, are specified using BPEL. A compiler takes the two BPEL specifications as the input and generates a multi-agent system as an ISPL (Interpreted Systems Programming Language) program. The composite services represented in ISPL are verified using a symbolic checker MCMAS (Model Checker for Multi-Agent Systems) tailored for verification of multi-agent systems.

The work by Yeung [91] addresses the verification of choreography-based Web services where the involved parties are not willing to expose their internal processes. In this scenario, the service choreography specifies the coordination from a global perspective and serves as a contract among the involved parties. The proposed approach is based on the Web Services Choreography Description Language (WS-CDL) and WS-BPEL. The abstract and/or executable processes of the involved parties expressed in WS-BPEL and the choreography model expressed in WS-CDL are transformed into Communicating Sequential Processes (CSP). CSP is a formal language for describing interactions in concurrent systems supported by the model checker of the FDR (Failures-Divergences Refinement). Yeung's approach [91] uses FDR to verify conformance to the choreography model.

Bentahar *et al.* [93] address the verification of composite Web services in respect to the properties of deadlock freedom, safety and reachability. They distinguish between *operational behavior* which describes the business logic by identifying system functions, and *control behavior* which identifies a sequence of actions that operational behavior should follow. *Operational behavior* is the model to be checked while *control behavior* refers to the properties that the model should satisfy such as deadlock freedom, safety and reachability. Operational behavior is modeled using an extended finite state machine and then transformed into Kripke model which is sequentially transformed into Symbolic Model Verifier (SMV) code. The properties from control behavior are checked against SMV code using NuSMV model checker.

Process models expressed in OWL can be verified using Semantic Web Rule Language (SWRL) and Semantic Query-Enhanced Web Rule Language (SQWRL). Valiente *et al.* [94] formalize the processes in terms of an ontology described in OWL and illustrate how SWRL rules can be used for model consistency checking and identification of breaches in service level agreements.

The verification of business process models, like the verification of Web services, often includes some kind of transformation into formalisms better suited for such analysis. Fahland *et al.* [95] investigated business models for soundness using three approaches: the checker LoLA, the Woflan tool and the WebSphere Business Modeler validation. Each of the three approaches required translation of the initial model represented in IBM WebSphere Business Modeler. The first approach, using checker LoLA, required transformation to Petri net models; the second approach, using the Woflan tool required initial transformation into Petri net and subsequently into workflow net; and the third approach, using the IBM WebSphere Business Modeler required translation into workflow graph.

Klai and Desel [96] also addressed verification of business process models in respect to soundness properties. As a model of the business system they used the Symbolic Observation Graph (SOG) which is Binary Decision Diagram (BDD) based abstraction of the behavior of a system. Soundness properties are translated from Petri net representation to Labeled Transition Systems (LTS) notation and then checked on the SOG model.

Examples of checking different aspects of business process models include: verification of models against compliance rules [97], verification of compliance with security requirements [98], and verification of semantic business models [99]. Morimoto [100] portrays a survey of formal verification for business process modeling.

5. Ontologies for Business Process Modeling

Although business process modeling notations are effective for representing business processes, these notations rarely provide formal semantics. For instance, BPMN provides powerful graphical representations of business processes that enable human users to model such processes; nevertheless, its lack of formalized semantics represents a challenge for automated queries and for comparison of existing models. To address this deficit, ontologies provide a way of formalizing semantics of business processes. Therefore, ontologies also facilitate the integration of business process and Web service views of the process. Web service semantics have been an active area of research and have been addressed by a number of technologies, including OWL-S, DAML-S, and WSMO, which were introduced in Subsection 3.2. Business process semantics play a crucial role in making use of Web services for the execution of business processes. Accordingly, this section focuses on ontologies for business process modeling.

5.1 Ontologies for business process modeling: an overview

A number of ontologies for business process modeling have been proposed [25,26,101-103]. Nevertheless, the main objectives of these ontologies vary to some extent, as illustrated in Table 11. For example, the main objective of PSL is to facilitate the exchange of process information among information systems, while the intent of oXPDL is to enable process analysis by querying and reasoning. These differences in objectives have resulted in different ontologies.

The Process Specification Language (PSL) [25,26] aims to facilitate the exchange of process information among information systems by providing a standard, neutral language which can serve as a language for translation. The foundation of PSL is the PSL-CORE, which includes only the basic primitives necessary to describe the process. In addition, PSL includes two types of *extensions* [26]: *core theories* introduce new primitive functions and relations, while *definitional extensions* use the terminology of the *core theories* to introduce new definitions.

The Business Process Modeling Ontology (BPMO) [101] has the main objective of modeling the business process at the semantic level. BPMO captures domain-independent organizational aspects and control-flow constructs of business notation, process interaction features from BPEL, and service descriptions and invocations from Semantic Web Services (SWS).

The General Process Ontology (GPO) [103,105] is part of the semantic annotation framework responsible for meta-model annotation. GPO provides a common conceptualization of the concepts used in different process modeling languages. To align heterogeneous process model representations, the modeling language constructs are annotated using a GPO.

The Business Process Modeling sub-Ontology (BPMsO) [102] and its counterpart Service Oriented Modeling sub-Ontology (SOMsO) have the objective of relating business process and Web service models by establishing a common terminology of the two domains and subsequently creating relationships between their elements manually. Delgado *et al.* [102] analyzed the variations among definitions of terms from different sources and attempted to establish comprehensive descriptions.

The Process Interchange Ontology (oXPDL) [104] explicitly models the complete semantics of the XML Process Definition Language (XPDL) [31] in a Web ontology language. Thus, oXPDL enables the automatic transformation of XPDL business process models to ontology languages and consequently enables querying and reasoning over business process models using standard ontology reasoners.

It is important to highlight the Onto-ITIL ontology [94] even though it is not an ontology representation for business process modeling. Nevertheless, Onto-ITIL facilitates the integration of business and information technologies by capturing the best practice guidelines for IT service management expressed in the Information Technology Infrastructure Library (ITIL). Specifically, Onto-ITIL formalizes the semantics of the ITIL in terms of an ontology defined in OWL language combined with Semantic Web Rule Language (SWRL) and Semantic Query-Enhanced Web Rule Language (SQWRL).

5.2 Ontologies for business process modeling: similarities and differences

Even though the objectives of ontologies for business process modeling are different, they possess a number of similarities. First and foremost, these ontologies, regardless of their objectives, have been attempting to establish a form of common ground among different business process representations. PSL aims to establish a language for information exchange among information systems, GPO provides a common conceptualization for different modeling languages, BPMO incorporates generic domain-independent constructs, and BPMsO attempts to standardize business process terminology. An exception is the oXPDL ontology, which deals with only one business process representation, specifically XPDL.

Although all the described ontologies, with the exception of oXPDL, attempt to provide a form of generalization, they provide specifications at different levels of granularity. A large-granularity, high-generalization representation is provided by GPO, which contains only about 18 generic concepts. Next are the ontologies of BPMO and BPMsO, which define more concepts, but they are still very generic. On the other end of the granularity scale is oXPDL, with 125 concepts [31]. Although PSL-CORE consists of only four classes [26], PSL extensions introduce extensive additional terminology.

Table 11. Ontologies for business process modeling

ONTOLOGY	MAIN OBJECTIVES
Process Specification Language (PSL) [25,26]	Facilitating the exchange of process information among information systems
Business Process Modeling Ontology (BPMO) [101]	Modeling business processes at the semantic level
General Process Ontology (GPO) [103]	Managing semantic heterogeneity
Business Process Modeling sub-Ontology (BPMsO) [102]	Relating business process and service models
XPDL-compliant process ontology (oXPDL) [104]	Enabling process analysis by querying and reasoning over multiple models

Another significant aspect of ontologies for business process modeling is how they handle the specification of domain-specific elements. The BPMO and GPO approaches use external, domain-specific ontologies to address domain-specific business concepts. This enables them to remain domain-independent even though they are capable of describing domain-specific concepts and processes. On the other hand, BPMsO and oXPDL do not address domain-specific components: BPMsO is concerned only with the most relevant generic concepts and oXPDL only with XPDL formalization. PSL captures domain-specific aspects using *extensions* which are constructed as domain-specific expansions of generic ontologies.

6. Challenges and Opportunities

A large number of languages and technologies have been used to express business processes and Web services. Specifically, these languages and technologies vary in their application domain, level of semantic formalization, modeling approach, level of industry adoption, and availability of supporting tools. The importance of bridging the gap between business process representations and Web services, as well as the significance of dealing with the representation heterogeneity issue, has been recognized by the research community. Accordingly, research efforts have resulted in numerous publications on transformations between different process representations, as shown in Table 2. A review of these publications led us to identify the following integration challenges:

- The existence of a large number of languages and technologies used for representing business process models and Web services makes the possible number of transformations immense, thereby imposing challenges on the transformation process.
- A wide variation in modeling approaches, even within representations of the same domain, raises obstacles for transformation [106].
- Representation languages and technologies have been designed for diverse domains, resulting in differences in information content. This leads to incomplete or inaccurate transformation outputs in which not all source information is represented in the output.

A number of proposed transformation approaches have been reviewed with regard to the pair(s) of representations involved, the transformation direction, the transformation approach used, the benefits offered, the role of ontologies, and the transformations trends. This analysis has revealed the following opportunities for future research into the integration of business process modeling and Web services:

- A generic approach is needed that will provide guidelines for transformations between representations that have not yet been attempted. Each of the proposed approaches reviewed in this paper, with the exception of that proposed by Vanderhaeghen *et al.* [87], addresses a specific pair of technologies. Because the number of technologies used in business process modeling and in Web services is large, addressing every pair independently would be impractical. Therefore, more studies investigating generic approaches to transformation between business process and Web service models are needed. Vanderhaeghen *et al.* [87] proposed a generic procedure for transformation between different business process representations; nevertheless, additional research is needed to evaluate its applicability to pairs of business process and Web service models.
- Mappings should be represented in a formalized way so that they can be read by computers as well as understood by humans. In particular, it is difficult to comprehend a transformation fully from a freeform textual description, especially when a large number of systems are involved. Moreover, the possibilities for reuse of the mappings represented in such a way are very limited. One way of formalizing mappings is through the use of ontologies, as proposed by Norton *et al.* [3]. In their work, Norton *et al.* used ontology mappings for a specific representation pair, BPMO and BPEL. However, further research is needed to explore the applicability of this approach to other pairs of representations between business processes models and Web service technologies.
- The number of proposed technologies for RESTful services is large; it includes WADL, POWDER, RIDDL, SAWSDL, SA-REST, and others, as illustrated in Table 1. Moreover, RESTful services are sometimes considered the *de facto* standard for service design [42]. However, RESTful services are underrepresented in the current transformation approaches; only two transformation studies involving RESTful technologies were encountered in the literature review. Hence, there is a need to explore more fully the integration of business process models and RESTful Web services.
- A common execution framework is required. The execution of described mappings is commonly considered to be an implementation issue and is therefore not included in research papers. Exceptions to this trend are the studies by Cabral and Domingue [79], Bordbar *et al.* [67], and Vanderhaeghen *et al.* [87], which use ATL rules, a SiTra framework, and XSLT rules respectively. Transformation can be considered as a two-step process, the first being mapping and the second being the execution of the transformation. Therefore, a comprehensive generic transformation

approach entails the definition of a common execution framework which would be capable of executing mappings represented in a formalized way.

- The semantics of the business process should be addressed in the integration efforts because they are crucial in using Web services for the execution of business processes. Business process representations differ greatly at a syntactic level, as well as at a semantic level. For Web services, semantics facilitate automated or semi-automated service discovery, composition, and orchestration. Consequently, a comprehensive integration solution must address semantic heterogeneity of different business process and Web service models.
- A decrease in the number of representations through standardization would reduce the number of transformations required. However, because the advantages and disadvantages of representations vary in different contexts, it is likely that a variety of different representations will remain in use.

A need for a generic approach can be closely related to other identified requirements/opportunities, including a need for formalized mappings between representations, a common execution framework, a decrease in the number of representations through standardization, and further investigation of the integration of business process models and RESTful Web services. Nevertheless, such mentioned requirements and opportunities can even be perceived as preconditions for achieving a generic transformation approach. For example, formalized mappings between representations are required for the achievement of a generic approach. Such formalized mappings can be achieved through ontologies, as proposed by Norton *et al.* [3]. However, further evaluation needs to be performed in order to assess the potential as well as the limitations of ontologies as a way of formalizing mappings involving various pairs of representations.

A common execution framework can be considered a part of a generic approach responsible for the execution of formalized mappings. When the mappings are formalized, an execution framework would be responsible to carry out the actual transformation from the source to the target representation as defined in the mappings. Thus, an execution framework along with the formalized mappings would be considered closely related components of a generic approach as: formalized mappings would govern the choice of the execution framework or the execution framework would govern the choice of the representation of mappings. We consider the method of formalizing mappings as essential for achieving a generic approach and at the same time immensely challenging as mappings between several representations need to be expressed. The complexity of formalized mappings is evident in a number of papers which focus on mappings itself for transformation purposes, including Shen *et al.* [72], Aslam *et al.* [70], Scicluna *et al.* [66], Le *et al.* [68], Weidlich *et al.* [74], Ziemann and Mendling [85], Mendling and Ziemann [75], and Paolucci *et al.* [76]. Therefore, we believe that the issue of formalized mappings should be addressed first. Next, an execution framework would be driven by the chosen representation of formalized mappings.

Another identified requirement/opportunity that impacts the achievement of a generic approach is the decrease in the number of representations through standardization. Although, it is to expect that a number of representations will remain in use, a decrease in the number of involved representations would facilitate the achievement of a generic approach.

It is important to point out that few studies have so far investigated transformations involving RESTful technologies. Hence, before attempting to design a generic transformation approach, efforts should be made to better understand the relation between business process models and RESTful Web services. It is imperative that a design of a possible generic transformation approach accommodates RESTful Web services since they are becoming the *de facto* standard for service design [42].

7. Conclusion

The major challenge in automating business process execution involves bridging the gap between a business view of the processes and an executable view of the processes which implement the business activities. The significance of integrating the business process and Web service models, as well as the need to deal with the heterogeneity of representations, has been recognized by the research community, resulting in a variety of transformation approaches involving different representations from the business process and Web service perspectives.

This paper focused on reviewing previous work on the integration of business process representations and Web service technologies with the following objectives: first, to provide a perspective on the domain by summarizing, organizing, and categorizing transformations, and second, to identify challenges and opportunities in the field of semantic integration of the business and executable views of processes.

A perspective on the domain is provided by analyzing different aspects of the proposed transformation approaches, the main ones being the transformation approach and the pair(s) of representations involved in the transformation.

The majority of the proposed transformation approaches deal with only one pair of representations, except for the work of Vanderhaeghen *et al.* [87], which proposed a generic transformation process. However, its major shortcoming is that it entails the need for XML intermediary representations of the source and target models. In addition, studies have typically focused on unidirectional transformations, with the exception of Norton *et al.* [3] study which addressed a bidirectional transformation between BPMO and BPEL. Moreover, the proposed approaches have not formalized the mapping representation; rather, most of them have described a mapping only as freeform text. As an exception, Norton *et al.* [3] used an ontology to represent the mapping: BPMO2sBPEL ontology represents the transformation from BPMO to BPEL, and BPEL2BPMO denotes the transformation from BPEL to BPMO.

Consequently, opportunities for future research in the domain include: designing a generic approach to transformation, formalizing representation of mappings, establishing a common execution framework, exploring the integration of business process models with RESTful Web services, addressing the semantics of the business processes in the integration and exploring the possibility of decreasing the number of representations through standardization.

APPENDIX

ACRONYMS AND ABBREVIATIONS

AML	- ARIS markup Language
ATL	- Atlas Transformation Language
BDD	- Binary Decision Diagram
BPD	- Business Process Diagrams
BPEL	- Business Process Execution Language
BPEL4WS	- BPEL for Web Services
BPM	- Business Process Management
BPML	- Business Process Modeling Language
BPMN	- Business Process Modeling Notation
BPMO	- Business Process Modeling Ontology
BPMsO	- The Business Process Modeling sub-Ontology
BPEL2BPMO	- BPEL to BPMO transformation
BPMO2sBPEL	- BPMO to sBPEL transformation
BWW	- Bunge-Wand-Weber
CSP	- Communicating Sequential Processes
DAML-S	- DARPA Agent Markup Language for Services
ebBPSS	- ebXML Business Process Specification Schema
EPC	- Event-Driven Process Chains
EPML	- EPC Markup Language
EXPRESS	- EXPressing REstful Semantic Services
FBPML	- Fundamental Business Process Modeling Language
FBPML PL	- FBPML Process Language
FDR	- Failures-Divergences Refinement
GPO	- General Process Ontology
hRESTS	- HTML for RESTful Services
IDEF3	- Integration Definition 3
ISPL	- Interpreted Systems Programming Language
ITIL	- Information Technology Infrastructure Library
KIBS	- Knowledge-intensive business services
MCMAS	- Model Checker for Multi-Agent Systems
OASIS	- Organization for Advancement of Structured Information Standards
OMG	- Object Management Group
Onto-ITIL	- ITIL Ontology
OWL	- Web Ontology Language
OWL-S	- OWL for Services
oXPDL	- Ontology for XPDL
POWDER	- Protocol for Web Description Resources
POWDER-S	- Semantic POWDER
PSL	- Process Specification Language
REST	- Representational State Transfer
RESTful	- Conforming to REST constraints
RIDDL	- RESTful Interface Definition and Declaration Language

SA-REST	- Semantic Annotations for SA-REST
SAWSDL	- Semantic Annotations for WSDL
sBPEL	- Semantic BPEL
SESE	- Single-entry Single-exit
SiTra	- Simple Transformer
SMV	- Symbolic Model Verifier
SOAP	- Simple Object Access Protocol
SOG	- Symbolic Observation Graph
SOMsO	- Services Oriented Modeling sub-Ontology
SQWRL	- Semantic Query-Enhanced Web Rule Language
SWRL	- Semantic Web Rule Language
SWSF	- Semantic Web Services Framework
UDDI	- Universal Description, Discovery, and Integration
WADL	- Web Application Description Language
WFCP	- Workflow Control Patterns
WS-BPEL	- Web Services BPEL
WS-CDL	- Web Services Choreography Description Language
WSDL	- Web Service Description Language
WSDL-S	- Web Service Semantics (WSDL Semantics)
WSML	- Web Service Modeling Language
WSMO	- Web Service Modeling Ontology
XML	- Extensible Markup Language
XPDL	- XML Process Definition Language
XSLT	- eXtendable Stylesheet Language Transformation
YAWL	- Yet Another Workflow Language

REFERENCES

1. Ouyang C, Dumas M, ter Hofstede AHM, van der Aalst WMP (2008) Pattern-based translation of BPMN process models to BPEL Web services. Int J Web Serv Res 5(1):42-62.
2. Nadarajan G, Chen-Burger Y-H (2007) Translating a typical business process modelling language to a Web services ontology through lightweight mapping. IET Softw 1(1):1-17.
3. Norton B, Cabral L, Nitzsche J (2009) Ontology-based translation of business process models. In: Proc Int Conf Internet Web Appl Serv, pp. 481-486
4. Girault C, Valk R (2003) Petri nets for systems engineering: a guide to modeling, verification, and applications. Springer, New York, NY
5. Milner R (1999) Communicating and mobile systems: the Pi calculus Cambridge University Press, Cambridge, UK
6. Arkin A (2002) BPML specification. http://www.bpmi.org/downloads/spec_down_bpml.htm
7. Object Management Group (OMG) (2010) Business process model and notation (BPMN). http://www.omg.org/spec/BPMN/
8. Dumas M, van der Aalst, W. M., ter Hofstede AH (2006) Process-aware information systems: bridging people and software through process technology. Wiley-Interscience, Hoboken, N.J
9. van der Aalst WMP, ter Hofstede AHM (2005) YAWL: yet another workflow language. Inf Syst 30(4):245-275.
10. Fielding R (2000) Architectural styles and the design of network-based software architectures, Chapter 5: Representational state transfer (REST). http://www.ics.uci.edu/~fielding/pubs/dissertation/rest_arch_style.htm
11. Richardson L, Ruby S (2007) RESTful Web services. O'Reilly Media, Portland, OR
12. Hadley M (2009) Web application description language, W3C submission. http://www.w3.org/Submission/wadl/
13. Andrews T, Curbera F, Dholakia H, Goland Y, Klein J, Leymann F, Liu K, Roller D, Smith D, Thatte S, Trickovic I, Weerawarana S (2003) Business process execution language for Web services 1.1. http://msdn.microsoft.com/en-US/library/ee251594%28v=bts.10%29.aspx
14. OASIS consortium (2007) Web services business process execution language, version 2.0. http://docs.oasis-open.org/wsbpel/2.0/OS/wsbpel-v2.0-OS.html
15. Booth D, Liu CK (2007) Web services description language (WSDL), version 2.0, W3C recommendation. http://www.w3.org/TR/2007/REC-wsdl20-primer-20070626/
16. Gudgin M, Hadley M, Mendelsohn N, Moreau JJ, Nielsen HF, Karmarkar A, Lafon Y (2007) SOAP, version 1.2, part 1: Messaging framework, W3C recommendation. http://www.w3.org/TR/soap12-part1/#intro
17. Fensel D, Facca FM, Simperl E, Toma I (2011) Semantic Web services. Springer, Berlin, Heidelberg
18. Studer R, Abecker A, Grimm S (2007) Semantic Web services: concepts, technologies, and applications. Springer, Berlin, Heidelberg
19. Studer R, Benjamins VR, Fensel D (1998) Knowledge engineering: principles and methods. Data Knowl Eng 25(1-2):161-197.
20. Dustdar S, Schreiner W (2005) A survey on Web services composition. Int J Web and Grid Services 1(1):1-30.
21. Rao J, Su X (2005) A survey of automated Web service composition methods. Springer, Berlin, Heidelberg, pp. 43-54
22. van der Aalst W, ter Hofstede A, Weske M (2003) Business process management: a survey. In: Proc Int Conf Bus Process Manag 2678, pp. 1-12
23. Mendling J, Nüttgens M (2008) Vorwort: XML integration and transformation for business process management (XML4BPM). In: Proc Multikonferenz Wirtschaftsinformatik
24. Lohmann N, Verbeek E, Dijkman R (2009) Petri net transformations for business processes – A survey. Transactions on petri nets and other models of concurrency II. , pp. 46-63
25. Schlenoff C, Gruninger M, Ciocoiu M, Lee J (1999) The essence of the process specification language. Trans Soc Comput Simul Int 16(4):204-216.
26. Gruninger M, Menzel C (2003) The process specification language (PSL): theory and applications. AI Mag 24(3):63-74.

27. Cheng J, Gruninger M, Sriram RD, Law KH (2003) Process specification language for project information exchange. Int J IT Archit Eng Constr 1(4):307-328.
28. Schlenoff C, Ciocoiu M, Libes D, Gruninger M (1999) Process specification language (PSL): results of the first pilot implementation. In: Proc Int Mech Eng Congr Expo
29. Papazoglou MP, Kratz B (2007) Web services technology in support of business transactions. Serv Comput Appl 1(1):51-63.
30. van der Aalst WMP, ter Hofstede AHM, Kiepuszewski B, Barros AP (2003) Workflow patterns. Distrib Parallel Databases 14(1):5-51.
31. Workflow Management Coalition (2008) XPDL, XML process definition language. http://www.wfmc.org/xpdl.html
32. Chen-Burger Y, Tate A, Robertson D (2002) Enterprise modelling: a declarative approach for FBPML. In: Proc Eur Conf Artif Intell Knowl Manag Organ Mem Workshop
33. Mayer RJ, Menzel CP, Painter MK, deWitte PS, Blinn T, Perakath B (1995) Information integration for concurrent engineering (IICE) IDEF3 process description capture method report. http://www.idef.com/pdf/Idef3_fn.pdf
34. OASIS Business Process Project Team (2001) ebXML business process specification schema, version 1.01. http://www.ebxml.org/specs/ebBPSS.pdf
35. van der Aalst WMP (2013) Business process management: a comprehensive survey. ISRN Softw Eng 2013
36. Ko RKL, Lee SSG, Lee EW (2009) Business process management (BPM) standards: a survey. Bus Process Manag J 15(5):744-791.
37. Eindhoven University of Technology.WofYAWL tool. http://www.win.tue.nl/woflan/doku.php?id=yawl
38. Ouyang C, Dumas M, ter Hofstede AHM, van der Aalst WMP (2006) From BPMN process models to BPEL Web services. In: Proc Int Conf Web Serv, pp. 285-292
39. Lee J, Kim K, Kim H, Kim C, Seo W (2011) Condition-based process patterns for modeling of human processes in knowledge-intensive business services. Expert Syst Appl 38(4):4025-4038.
40. Tang L, Dong J, Peng T, Tsai W (2010) Modeling enterprise service-oriented architectural styles. Serv Comput Appl 4(2):81-107.
41. Khorasgani RR, Stroulia E, Zaiane OR (2011) Web service matching for RESTful Web services. In: Proc 13th IEEE Int Symp Web Syst Evol, pp. 115-124
42. Battle R, Benson E (2008) Bridging the semantic Web and Web 2.0 with representational state transfer (REST). Web Semant: Sci Serv Agents World Wide Web 6(1):61-69.
43. Archer P, Smith K, Perego A (2009) Protocol for Web description resources (POWDER): description resources, W3C recommendation. http://www.w3.org/TR/powder-dr/
44. Mangler J, Schikuta E, Witzany C (2009) Quo vadis interface definition languages? Towards a interface definition language for RESTful services. In: Proc IEEE Int Conf Serv Oriented Comput Appl
45. Kopecký J, Gomadam K, Vitvar T (2008) hRESTS: an HTML microformat for describing RESTful Web services. In: Proc IEEE/WIC/ACM Int Conf Web Intell Intell Agent Technol
46. Davies J, Studer R, Warren P (2006) Semantic Web technologies: trends and research in ontology-based systems. John Wiley & Sons, Hoboken, NJ
47. Berners-Lee T, Fischetti M (1999) Weaving the web: the original design and ultimate destiny of the World Wide Web by its inventor. Harper, San Francisco, CA
48. Farrell J, Lausen H (2007) Semantic annotations for WSDL and XML schema. http://www.w3.org/TR/sawsdl/
49. Akkiraju R, Farrell J, Miller JA, Nagarajan M, Sheth A, Verma K (2006) Web service semantics WSDL-S, IBM research report. http://domino.research.ibm.com/library/cyberdig.nsf/0/ef9fe52551fb21dc8525710d005a8480?OpenDocument
50. Pautasso C (2008) BPEL for REST. Springer, Berlin, Heidelberg, pp. 278-293
51. Pautasso C (2009) RESTful Web service composition with BPEL for REST. Data Knowl Eng 68(9):851-866.
52. Ankolekar A, Burstein M, Hobbs JR, Lassila O, Martin D, McDermott D, McIlraith SA (2002) DAML-S: Web service description for the semantic web. In: Proc First Int Conf Web Services, pp. 348-363
53. Martin D, Burstein M, Hobbs J, Lassila O, McDermott D, McIlraith S, Narayanan S, Paolucci M, Parsia B, Payne T, Sirin E, Srinivasan N, Sycara K (2004) OWL-S: semantic markup for Web services, W3C submission. www.w3.org/Submission/OWL-S/
54. de Bruijn J, Bussler C, Domingue J, Fensel D, Hepp M, Keller U, Kifer M (2005) Web service modeling ontology (WSMO), W3C submission. http://www.w3.org/Submission/WSMO/
55. Roman D, Keller U, Lausen H, de Bruijn J, Lara R, Stollberg M, Polleres A, Feier C, Bussler C, Fensel D (2005) Web service modeling ontology. Appl Ontol 15(1):77-106.
56. Feier C, Roman D, Polleres A, Domingue J, Stollberg M, Fense D (2005) Towards intelligent Web services: Web service modeling ontology. In: Proc Int Conf Intell Comput
57. ESSI WSML Working Group (2008) The Web service modeling language WSML specification. http://www.wsmo.org/wsml/wsml-syntax
58. SWSL Committee (2005) Semantic Web services framework (SWSF). http://www.daml.org/services/swsf/1.0/
59. Gomadam K, Ranabahu A, Sheth A (2010) SA-REST: semantic annotation of Web resources, W3C submission http://www.w3.org/Submission/SA-REST/
60. Perego A, Konstantopoulos S, Archer P (2010) Protocol for Web description resources (POWDER): POWDER-S vocabulary (WDRS). http://www.w3.org/2007/05/powder-s
61. Alowisheq A, Millard DE, Tiropanis T (2009) EXPRESS: EXPressing REstful semantic services using domain ontologies. In: Proc 8th Int Semant Web Conf
62. Lara R, Polleres A, Lausen H, Roman D, de Bruijn J, Fensel D (2005) A conceptual comparison between WSMO and OWL-S. http://www.wsmo.org/2004/d4/d4.1/v0.1/20050106/d4.1v0.1_20050106.pdf
63. de Bruijn J, Fensel D, Kifer M, Kopeck J, Lara R, Lausen H, Polleres A, Roman D, Scicluna J, Toma I (2005) Relationship of WSMO to other relevant technologies, W3C submission. http://www.w3.org/Submission/WSMO-related/
64. Guo L (2010) B2B collaboration through Web services-based multi-agent system. Serv Comput Appl 4(2):137-153.
65. Mendling J, Nüttgens M (2004) Transformation of ARIS markup language to EPML. In: Proc 3rd GI Workshop on Event-Driven Process Chains, pp. 27-38
66. Scicluna J, Lara R, Polleres A, Lausen H (2004) Formal mapping and tool to OWL-S. WSMO Working Group Report
67. Bordbar B, Howells G, Evans M, Staikopoulos A (2007) Model transformation from OWL-S to BPEL via SiTra. In: Proc Third Eur Conf Model-Driven Archit: Found Appl, pp. 43-58
68. Le D, Ong CJD, Goh AES (2009) Converting WSMO to OWL-S system. In: Proc IEEE Asia-Pacific Services Comput Conf, pp. 7-12
69. Shen J, Yang Y, Zhu C, Wan C (2005) From BPEL4WS to OWL-S: Integrating e-business process descriptions. In: Proc IEEE Int Conf Services Comput 1, pp. 181-188
70. Aslam MA, Auer S, Shen J, Herrmann M (2006) Expressing business process models as OWL-S ontologies. In: Proc Bus Process Manag Workshops, pp. 400-415

71. Wang S, Capretz MAM, El Yamany HF (2009) Semantic profile based service discovery for abstract process composition. In: Proc Int Conf e-Business, pp. 258-263
72. Shen J, Yang Y, Lalwani B (2004) Mapping Web services specifications to process ontology: opportunities and limitations. In: Proc 10th IEEE Int Workshop Future Trends Distrib Comput Syst, pp. 235-235
73. Brogi A, Popescu R (2006) From BPEL processes to YAWL workflows. In: Proc Third Int Workshop Web Serv Form Meth, pp. 107-122
74. Weidlich M, Decker G, Grosskopf A, Weske M (2008) BPEL to BPMN: the myth of a straight-forward mapping. In: Proc OTM Confed Int Conf Move Meaningful Internet Syst, pp. 265-282
75. Mendling J, Ziemann J (2005) Transformation of BPEL processes to EPCs. In: Proc 4th GI Workshop Event-Driven Process Chains
76. Paolucci M, Srinivasan N, Sycara K, Nishimura T (2003) Toward a semantic choreography of Web services: from WSDL to DAML-S. In: Proc First Int Conf Web Services, pp. 22-26
77. Upadhyaya B, Zou Y, Xiao H, Ng J, Lau A (2011) Migration of SOAP-based services to RESTful services. In: Proc 13th IEEE Int Symp Web Syst Evol (WSE), pp. 105-114
78. Peng Y, Ma S, Lee J (2009) REST2SOAP: a framework to integrate SOAP services and RESTful services. In: Proc IEEE Int Conf Service-Oriented Comput Appl
79. Cabral L, Domingue J (2009) Translating semantic Web service based business process models. In: Proc IEEE Asia-Pacific Serv Comput Conf, pp. 1-6
80. García-Bañuelos L (2008) Pattern identification and classification in the translation from BPMN to BPEL. In: Proc Confed Int Conf Move Meaningful Internet Syst, pp. 436-444
81. White SA (2005) Using BPMN to model a BPEL process. http://www.bpmn.org/Documents/Mapping_BPMN_to_BPEL_Example.pdf
82. Ye J, Sun S, Wen L, Song W (2008) Transformation of BPMN to YAWL. In: Proc Int Conf Comput Sci Softw Eng 2, pp. 354-359
83. Decker G, Dijkman R, Dumas M, García-Bañuelos L (2008) Transforming BPMN diagrams into YAWL nets. In: Proc 6th Int Conf Bus Process Manag, pp. 386-389
84. Guo L, Chen-Burger Y, Robertson D (2004) Mapping a business process model to a semantic Web service model. In: Proc IEEE Int Conf Web Serv, pp. 746-749
85. Ziemann J, Mendling J (2005) EPC-based modelling of BPEL processes: a pragmatic transformation approach. In: Proc 13th Int Conf Mod Inform Technol Innov Processes Ind Enterp
86. Meertens LO, Iacob M, Eckartz SM (2010) Feasibility of EPC to BPEL model transformations based on ontology and patterns. Lect Notes Bus Inform Process 43:347-358.
87. Vanderhaeghen D, Zang S, Hofer A, Adam O (2005) XML-based transformation of business process models – enabler for collaborative business process management. In: Proc Second GI Workshop (XML4BPM)
88. Ko RKL (2009) A computer Scientist's introductory guide to business process management (BPM). ACM Crossroads 15(4):4-18.
89. Wand Y, Weber R (1990) An ontological model of an information system. IEEE Trans Softw Eng 16(11):1282-1292.
90. Lomuscio A, Qu H, Solanki M (2012) Towards verifying contract regulated service composition. Auton Agents Multi-Agent Syst 24(3):345-373.
91. Yeung WL (2011) A formal and visual modeling approach to choreography based Web services composition and conformance verification. Expert Syst Appl 38(10):12772-12785.
92. Hull R, Su J (2005) Tools for composite Web services: a short overview. ACM SIGMOD Record 34(2):86-95.
93. Bentahar J, Yahyaoui H, Kova M, Maamar Z (2013) Symbolic model checking composite Web services using operational and control behaviors. Expert Syst Appl 40(2):508-522.
94. Valiente M, Garcia-Barriocanal E, Sicilia M (2012) Applying an ontology approach to IT service management for business-IT integration. Knowledge-Based Syst 28:76-87.
95. Fahland D, Favre C, Koehler J, Lohmann N, Völzer H, Wolf K (2011) Analysis on demand: instantaneous soundness checking of industrial business process models. Data Knowl Eng 70(5):448-466.
96. Klai K, Desel J (2012) Checking soundness of business processes compositionally using symbolic observation graphs. Formal Techniques for Distributed Systems 7273(Form Tech for Distrib Syst):67-83.
97. Knuplesch D, Ly LT, Rinderle-Ma S, Pfeifer H, Dadam P (2010) On enabling data-aware compliance checking of business process models. In: Proc 29th Int Conf on Conceptual Model 6412, pp. 332-346
98. Armando A, Ponta SE (2010) Model checking of security-sensitive business processes. Formal aspects in security and trust. Springer, Berlin, Heidelberg, pp. 66-80
99. Weber I, Hoffmann J, Mendling J (2010) Beyond soundness: on the verification of semantic business process models. Distrib Parallel Databases 27(3):271-343.
100. Morimoto S (2008) A survey of formal verification for business process modeling. Computational science. Springer, Berlin, Heidelberg, pp. 514-522
101. Cabral L, Norton B, Domingue J (2009) The business process modelling ontology. In: Workshop: Semant Bus Process Manag (SBPM 2009) at Eur Semant Web Conf (ESWC 2009)
102. Delgado A, Ruiz F, de Guzmán IGR, Piattini M (2010) Towards an ontology for service oriented modeling supporting business processes. In: Proc Fourth Int Conf Res Chall Infor Sci (RCIS), pp. 415-424
103. Lin Y, Strasunskas D, Hakkarainen S, Krogstie J, Solvberg A (2006) Semantic annotation framework to manage semantic heterogeneity of process models. Advanced information systems engineering. Springer, Berlin, Heidelberg, pp. 433-446
104. Haller A, Gaaloul W, Marmolowski M (2008) Towards an XPDL compliant process ontology. In: Proc IEEE Congr Serv - Part I, pp. 83-86
105. Lin Y, Ding H (2005) Ontology-based semantic annotation for semantic interoperability of process models. In: Proc Int Conf Comput Intell Model Control Automat
106. Mendling J, Lassen KB, Zdun U (2008) On the transformation of control flow between block-oriented and graph-oriented process modeling languages. Int J Bus Process Integr Manag 3(2):96-108.