Business Process Model Transformation Techniques: A Comprehensive Survey

Ahsanun Naseh Khudori, Tri Astoto Kurniawan
Software Engineering Research Group, Faculty of Computer Science
Brawijaya University, Malang – Indonesia.
ahsanunnaseh.bimasakti@gmail.com

***Abstract*— Business Process Modeling (BPM) plays a critical role in improving organizational efficiency in the industry. It is important to documenting, analyzing and optimizing workflows. BPM is frequently applied on different levels of abstractions and formality. There are various BPM technologies and languages including BPMN, EPC. The enterprise is always changing and evolving that may have impacts on business processes models, which are modeled in various BPM languages. As such, these models have to be changed. Such changes may involve process model transformation between different languages. Consequently, it requires a proven technique for such model transformation. There exist various process model transformations with different proposes. In this paper, we are interested in performing a literature survey on the existing business process transformation techniques. We reviewed 23 techniques using 3 criteria, i.e. completeness, correctness and tool support, in order to compare such techniques. The result indicates that the existing techniques have not completely transformed each process-modeling notation between different languages.**

***Index Terms*— Model Transformation Techniques; Business Process Modelling; Survey, Comparison**

I. INTRODUCTION

The last few decades, BPM has become an essential component of the enterprise management [1]. it is one of the international standard requirements for quality management and assurance (ISO 9000) and one of the key questions when applied the enterprise systems, as electronic business, resources planning of business and modeling language for operational standards of enterprise [2]. BPM plays the critical role for improving organizational efficiency in the industry. It is important to documenting, analyzing and optimizing workflows. A process model is critical to the success for applying information systems development and business process improvement. In improving competitiveness, business process improvement is a very effective way to make changes in function and structure of organizational. BPM is a key element to analyze the effectiveness of the business processes of an organization [2].

There are a variety of languages, methods and techniques that implemented the problem of BPM. Such as flowchart technique proposed by American National Standards Institute (ANSI), Unified Modeling Language (UML) technique that proposed by Fowler to use as part of software design, Event-driven Process Chains (EPC) proposed by Scheer dedicated to business-oriented modeling approaches, PN proposed by Petri based on formalized and academically studied techniques. And the latest is BPM Notation (BPMN) proposed by OMG with the goal to provide a notation that is easily readable and understandable. Since the change of business for many years, such as business incorporation or new inventions in the organization needs new model business communication. So, they need to improve and remodeling existing process business to synchronize or translate into a current Business Process Model Language (BPML).

The enterprise is always changing and developing that may have impacts on BPM, which a modeled in various BPML. Such changes may involve process model transformation between different languages. Consequently, it's requires a proven technique for such model transformation. Moreover, enterprises need a model transformation technique of business process models to conserve the semantics of a model [3].

Model transformation techniques for BPM have achieved a certain level of maturity [4]. These have been transform from a BPM to another, such as EPC to BPMN [1], [5], [6], [7] Petri Nets (PN) to BPMN [8], [9] [10], [11], [12], BPMN to Yet Another Workflow Language (YAWL) [13], [14], BPMN to DEVS [15], [16], [17], BPMN to UML [18], [19], [20] and other. However, defining the transformation in several areas of application in the case of the BPM remains a challenge because the existing transformation language only gives a common solution. Reference [21] have been discussed 7 model transformation issues, i.e. *decision (un)ambiguity, invisible merger, mandatory events, different start objects, split/merge unambiguity, join specification problem and different final nodes.* The aim of this paper is to review and compare the model transformation techniques on BPM. The trend is the use of BPM is growing rapidly and many of the proposed methods to transform between different BPM justify the importance of a comprehensive survey.

Specifically, this paper has following objectives: to integrate work in the domain and to provide a perspective on model transformation techniques by summarizing and categorizing transformation between BPM representations. Such a categorization will help practitioners locate studies relevant to a specific problem, while a perspective on the model transformation will help researchers identify research directions, research challenges, opportunities in the field of model transformation approaches. In other, each model transformation has been proposed based on their own

objectives and approaches. So, there is no standard in assessing model transformation techniques. It is not easy for a user community to determine to choose a proper model transformation technique. Further, there has been no previous studies to evaluate to what extent distinguish exist between model transformation techniques. In order to provide it, this work is intended to deal with a comparison of BPM transformation techniques, which relies on graphical BPM. The result of this study may help researchers or practitioners to determine their relevant model transformation technique.

This paper present the related works in section 2, while the next section introduces basic concepts associated with BPM and models transformation. In Section 4 describes review process, how authors select model transformation literature and their criteria. The finding and discussion of various transformation approaches technique between different BPM are reviewed in the Section 5, and the conclusion presented in the last section, which summarizes this work and provide insight for future research.

II. RELATED WORKS

There is a little study to review the model transformation techniques of BPM. Reference [22] give the abstract of a concrete transformation by differentiating two main paradigms to represent the control flow in the BPM: block-oriented language, as BPML and Business Process Execution Language (BPEL), and graph-oriented language, as EPC, YAWL and BPMN. Their contribution is a generic strategy to transform from block-oriented to graph-oriented languages, and otherwise. Reference [23] discussed a number of transformations on Petri networks that already exist; they analyzed the transformation of PN from another BPM. In other, they investigated the lacks of conceptual transformations, algorithm and tools are used to transform. Reference [24] conducted a review on transformation between BPM and web services. They provided a perspective on the field to summarize and organize transformation approaches of BPM, BPM to web services and between web services.

Our study focuses on model transformation between BPM to BPM, especially graphical languages category. In other, we also propose a framework to compare model transformation techniques approach used by researchers.

III. BPM AND MODELS TRANSFORMATION

A. Business Process Modeling

BPM generates an artifact of basic conceptual management of organizational processes and sustainable change [24]. It is an activity that represents the company's business processes so that it can be understood, analyzed, corrected and improved. In addition, it also includes the activities of design, modeling, execution, monitoring and optimization.

There are many BPM languages and techniques including:

BPMN describe business processes in the form of graphical representation [25], the main concept is similar to Activity Diagram (AD). In addition, designed for BPM, BPMN is also designed to change BPML.

EPC, describe business processes with a flowchart-style representation [26]. EPC developed with the goal of keeping the business processes easily understood and used. EPC have two basic elements namely functions and events. [27].

UML Activity Diagram (UML AD), describe business processes using the fork node that generates a set of parallel paths [21]. UML AD including one suitable modeling language used to describe the interactions between objects of the system so that it can understand the business processes, workflow and interaction scenarios with ease [28].

YAWL: is a language of workflow that refers to the patterns of workflow [29]. It is language based on state and specifications of semantics workflow is defined as a transition system [14]. YAWL supports workflow concise language as well as complex and powerful, and can handle complex data transformations.

Petri Net, a design for dynamic systems modeling, analysis, and simulation that based on the procedures of concurrent and nondeterministic, which is a directed graph [9], which is a directed graph. PN is a language modeling that used to modeling workflow that consists primarily of two different nodes i.e. transitions and places [30].

B. Business Process Models Transformation

Models transformation is an important key technique for automatic of modeling artifacts of workflow management system [31]. In the last years, several of approaches, technology, and languages on models transformation have been explored. Reference [32] define the model transformation basic concepts as follows:

- A source model associated with the given source meta-model (model structures).
- A target model associated with the given target meta-model (model structures).
- Transformation rules based on the source and target meta-model (model structures).
- Transformation engine, it produces a target model from source model by read a source model and executes a transformation technique.

Model transformation technique can be classified by two levels abstraction, i.e. high-level and low-level abstraction. High-level defines the meta-model and transformation rules, which describe the mapping between different models, while low-level instantiated a source, target and engine of transformation that executes the rules to change the model [33]. Besides that, there are two areas of model transformation i.e. horizontal and vertical model transformation. Horizontal model is designed when migrating between different modeling languages at the same level, while vertical model is designed when moving from platform independent models to platform dependent model or even to code [34]. In other there are various approaches of model transformation, including Extensible Style sheet Language Transformations (XSLT), Atlas Transformation Language (ATL) [36], and Query/Views/Transformations (QVT). XSLT in [35] is a technology of XML transformation which is the first promising candidate in to realize model transformations, it is very fulfilling to a simple transformation, but it has serious drawbacks for the more advanced transformations. QVT in [36] is developed by the OMG with the goal to standardize model transformations; it defines four different specifying

model transformations ways. The Atlas Group and the TNI-Valiosys Company explores ATL, which purposed facilitates the practical implementation of the QVT [37]

IV. METHODOLOGY

A. Selection of Models Transformation

Reference [38] proposes BPM Language into 3 clear-cut types of standards. *First,* graphical language intended to describe the business process and the possibility of flow and transition in the form of a diagram. *Second*, execution language used for develops a computerization and automation of business process. *Third* interchange standard is aimed for facilitating data probability, including of the business process designs portability in diverse graphical standards within Business Process Management System (BPMS), disparate execution standards within different BPMS, as soon as translation of context-less than the graphics standard for execution standard and conversely.

Table 1
Business Process Model Categories in [38]

BPM Standard	Theory/ graphical/ interchange/ execution	Standardized	Status
EPC	Graphical	Yes	Stable
BPMN	Graphical	Yes	Popular
Flowchart	Graphical	NA	Popular
UML-AD	Graphical	Yes	Popular
RAD	Graphical	Yes	NA
YAWL	Graphical/ Execution	No	Stable
Petri-nets	Theory/ Graphical	NA	Popular
BPML	Execution	Yes	Obsolete
BPEL	Execution	Yes	Popular
XLANG	Execution	NA	-
WSFL	Execution	No	Obsolete
Pi-Calculus	Execution	NA	Popular
BPEL4WS/ WS-BPEL	Execution	Yes	Popular
BPDM	Interchange		NA
XPDL	Execution/ Interchange	Yes	Stable
BPMD	Interchange	Yes	NA

But, reference [29] distinguishes the BPM into 3 categories: formal languages, conceptual languages, and execution languages. Formal languages are referenced on theoretical models. In common, these languages have unambiguous semantic and allow for analysis. In practice, users have the problem using this formal language [29]. 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. Execution languages such as BPEL typically abstract from implementation details. While reference [29] 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. In practice, users often prefer conceptual languages to formal languages due to the ease of use and informality [29].

This research focuses on graphical modeling languages. The motive for choice the graphical languages techniques are ordinarily used by researchers or practitioners to describing and modeling an environment, both people, and organizations, of a computerized or automation system in representing problematic and complicated situations in organizations [39]. EPC, BPMN, UML-AD, and flowchart are the frequently used in graphical languages. Another example of graphical modeling language that gets a lot of interest in the research community is a Petri Net and YAWL.

B. Paper Inclusion Criteria and Identification

A paper selected to be included in this review meets the following criteria: it describes a transformation between a different BPM, graphical languages. An initial group of papers was identified by searching the ScienceDirect, Springer Link, and IEEE Xplored databases.

Searches were performed using combinations of the term "transform" such as search phrases were of the form "BPMN transform", "EPC transform", "YAWL transform", "Petri Net transform", "Flowchart transform", "UML AD Transform", and "RAD transform". From the search results, authors defined relevant papers by reading the title and continued by reading abstract papers if the title was not sufficient.

In an attempt to identify additional papers on the transformation between different business processes modeling, Google scholar searches were performed using the same strategy of combining the term "transform" and the abbreviations of technologies. In spite of the efforts to conduct a comprehensive search, it is possible that some papers have been missed. The search was completed in January 2017.

C. Comparison Criteria

This study wanted to compare model transformation approaches in graphical model transformation scenario and analyze whether there are any differences between that approaches. Moreover, the author wants to stimulate in-depth discussion on the approach of the transformation model, and be a trigger of research agenda that aims to integrate the strengths of all approaches. So that, to facilitate comparison of the model transformation approaches, we propose solutions to the assessment criteria in Table 2.

Table 2
Comparison Criteria

Criteria	Description
Completeness	All notations in one model should be completely transformed to another model.
Correctness	The execution semantics of the business process model should be preserved by transformation. In addition, the transformation should be facilitated the argumentation about the correctness of translate and migration process.
Tool Support	The tool to transform is available and users can be used as well as which users have access to those tools.

To produce a more objective comparison, Authors divide criteria as mentioned in Table 2 into 3 level of assessment: High, Average, and Low as described in Table 3. Each BPM has a core of notation, that elements basic of process models. Core notation of EPC is a function, event, control flow, AND, XOR and OR connector. While in BPMN are the task, start and end event, sequence flow, data based exclusive, parallel

gateway, and inclusive gateway. Petri Nets core notations are transition, place, place with token, inflow, outflow, reset and inhibitor Arc. UML AD core notations are action, edge, fork, join, merge, decision, initial and final node. And YAWL core notations are input and output condition, remove tokens, composite task and multiple instances, condition, atomic, AND-split, XOR-split, OR-split, AND-join, XOR-join, and OR-join task.

Table 3
Comparison Criteria

Level	Completeness	Correctness	Tool Support
High	If all the notations of source model can be mapped into the target model.	The researcher presented complex study that involving all notations in source model and equivalence model can be checked on the corresponding target models.	If the tool can be accessed and can be used to transform from defined source model into the target model.
Average	If proposed mapping rules, only map the core of notation source model into the target models.	Researchers presented case study, using the core of notation source model and equivalence model can be checked on the corresponding target models.	If tool presented on the paper and authors presented link of the tool. But tool can't be accessed.
Low	If proposed mapping rules, not all core of notation source model can be mapped into the target models.	Researchers didn't present case study and researchers didn't have scenarios to examine proposed model	If researchers just proposed a prototype of the application to transform from source model into target models.

D. Threats to Validity

The main threats to the validity of this study are: *Keyword bias*, only papers written in English and full filled keyword criteria have been included. *Decision Selection bias*, the search for papers decision in focused on finding research papers, including journal and conference. As a result, it is possible relevant technical reports and white papers may have been excluded. *Unpublished transformation technique studies*, applying a known approach or writing code to make software transformation 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.

V. FINDING AND DISCUSSION

A. BPM Transformation Studies and Approaches

This paper focuses on model transformation between graphical BPM. Nevertheless, the search was also conducted for the model transformation to another categories language, execution languages and interchange standard languages. Relevant literature was obtained using search scenario described before presents and overview in Table 4. There were 59 papers identified.

In Table 4, in addition, to distinguishing between BPM based on [38] categories, so that paper will discuss by class of categories. The model transformation in graphical process model has more research attention than other categories. There were 19 papers that have been discussed in this field. The transformation from graphical languages to execution languages were 17 papers that have been discussed and the transformation from graphical languages to interchange standard, there were 9 papers that have been discussed.

Table 4
Transformation Between BPM

Busines Process Modeling (BPM) Standart			Transformation To								
			Graphical Languages					Execution Languages		Interchanges standards	
			EPC	BPMN	UML-AD	YAWL	Petri Net	BPEL	Pi-Calculus	WS-BPEL	XPDL
Transformation From	Graphical Languages	EPC	-	[1], [26]	-	-	-	[40], [41], [30]	-	-	-
		BPMN	[1]	-	[19]	[13], [42], [14], [43]	[12], [44], [10], [8], [11]	[45], [46], [47], [48] [49], [50], [51], [52], [53], [54], [55]	[56]	[57], [58], [59]	[60], [61]
		UML-AD	-	-	-	[62], [63]	[64], [65], [66], [67], [68]	-	-	[69], [70]	[71], [72]
		YAWL	-	[42]	-	-	-	[73]	-	-	-
		Petri-nets	[74]	-	-	-	-	[23]	-	-	
	Execution Languages	BPEL	[40]	[75], [50], [51]		[76]	[77]	-	-	-	-
		Pi-Calculus	-	-	-	-	-	-	-	-	-
	Interchanges standards	WS-BPEL	-	-	-	-	-	-	-	-	-
		XPDL	-	-	-	-	[78]	-	-	-	-

Transformation studies from table 5 were further examined with regard to the transformation direction and approaches that have been used. Each transformation has approach each other's; it is very important to note that each proposed approach address a specific pair of technologies. Especially, the researchers addressing in the opposite transformation. BPMN to PN and PN to UML AD have more attention than other categories. The full results are presented in table 4.

In the case of one publication that discussed the bidirectional transformation, it will be presented them in different publications. Such as [1] discussed bidirectional transformation, from BPMN to EPC and EPC to BPMN in one paper publication. Yee et al have studied BPMN to YAWL in 3 publication papers. The first paper discussed bidirectional transformation BPMN to YAWL and YAWL to BPMN. But in 2 publication papers focused on one directional transformation. Moreover, several transformations such as PN to BPMN, UML AD to BPMN, EPC to PN, UML AD to PN and YAWL to UML AD didn't have an appropriate opposite transformation.

Table 5
Model Transformation Approaches by Direction

Transformation in One Direction			Transformation in The Opposite Direction		
Transformation Direction	Study	Approach/ Tool	Transformation direction	Study	Approach/ Tool
EPC to BPMN	[1]	Direct Mapping Rule/ BPGen	BPMN to EPC	[1]	Conversion Rule/ BPGen
	[26]	Direct Mapping Rule and Semantic Rules/ Plugin in the Oryx-Editor			
Petri Net to BPMN	-		BPMN to PN	[8]	Mapping Rule/ NA
				[12]	Mapping Rule (Theoritical) / NA
				[11]	Atlas Transformation Language (ATL)/ NA
				[10]	Mapping Rule/ http://is.tm.tue.nl/staff rdijkman/cbd.html#transformer.
				[44]	ATL/ Time Petri Net Analyzer (TINA)
BPMN to YAWL	[13]	Mapping Rule/ Eclipse Plugin	YAWL to BPMN	[42]	Mapping Rule/ BPMN2YAWL Plugin in ProM 5.0
	[42] [43] [14]	Mapping Rule/ BPMN2YAWL Plugin in ProM 5.0			
BPMN to UML AD	[19]	XSLT/ NA	UML AD to BPMN	-	
PN to EPCs	[74]	Verbeek Translation Rule/ WF-Net Plugin in ProM	EPC to PN	-	
PN to UML AD	[64]	AD-CPNI Translation Rule/ NA	UML AD to PN	-	
	[65]	Intuitive Mapping Rule/ NA			
	[67]	Transformation Rule/ NA			
	[68]	Mapping Rule/ Prototype Tool			
UML AD to YAWL	[62] [63]	Mapping Rule/ NA	YAWL to UML AD	-	

B. Comparison of Models Transformation

The result of comparison of models transformation in BPM by 3 criteria and 3 levels assessment is presented in table 5 this bellow. The most mature graphical language models transformation proposed by [26] in EPC to BPMN, [10] in BPMN to Petri Net, and [62] in UML AD to YAWL. They implemented the transformation concept as plugin in software modeling tools, but cannot be accessed. While the lowest mature concept proposed by [64]. Discussion of the comparison results is presented for each transformation model to others model and vice versa.

Table 6
Comparison of Models Transformation

Model Transformation	Completeness	Correctness	Tool Support
BPMN-EPC-BPMN			
Kotsev et al [1]	**	**	**
Tscheschner [26]	***	***	**
BPMN-PN			
Raedts et. al [8]	**	**	*
Ramadan et al [12]	***	***	*
Kasar and Paliwal [11]	**	**	*
Dijkman et al [10]	***	***	**
Lyazidi and Mouline [44]	***	**	**
YAWL-BPMN-YAWL			
Decker et al [13]	**	**	***
Ye et al [14][42][43]	**	**	***
BPMN-UML AD			
Macek and Richta [19]	**	**	*
PN-EPC			
Verbeek and Dongen [74]	**	**	**
PN-UML AD			
Maneerat and Vatanawood [64]	**	*	*
Staines [65]	**	**	*
Lopez et al [66]	**	**	**
Trickovie [67]	**	**	*
Hu and shatz [68]	**	**	*
UML AD-YAWL			
Han et al [62] [63]	***	***	*

* : Low ** : Average *** : High

BPMN-EPC-BPMN. There were 2 papers in the field of transformation BPMN to EPC vice versa. Reference [26] Proposed a mapping rules entire notation in EPC to BPMN. But not every notation can ben transformed into BPMN notations caused difference notations between BPMN and EPC. He added semantic rules and some extensions to handle message event. Every detected "sent" function in the EPC can be translated to a "throwing" intermediate message and every detected "receiving" event can be translated to a "catching" intermediate message. He has defined 3 groups transformation rules from 9 rules, i.e. transformation core EPC, transformation extended EPC and transformation extension with semantic rules. Reference [26] have proposed the mature concept and implemented his concept as a plugin in the Oryx-Editor, But his tools can't be accessed. So, Reference [1] implemented Tsechesner mapping rules to develop BPMN-

EPC-BPMN converter tool. But just transform from the among core notations of EPC and BPMN as mentioned in comparison criteria section. They focused on the software architecture of the Business Process Generator (BPGen) to convert existing EPC model to BPMN model, and vice versa. This tool has 3 main parts single environment development, the core tool and the component BP-Gen Converter.

BPMN-PN. There were not studies in opposite transformation PN to BPMN. Transformation BPMN to PN has more attention. Raedts et al proposed mapping rules by proposing an approach to transforming original BPMN model automatically to PN Model. Subsequently, PN model can be transformed to mCRL2 using Petri net-based analysis tool, to verify and validate target model by using mCRL2 toolset. But not all notations in BPMN could be transforming to PN notation. Sub process, pool, data object, annotation, association, message notations not described specifically how to transform them into PN notations. In other, [11] used ATL approach, a hybrid model transformation language. They used mapping rules, which have described by [10]. BPMN Notations mapped to corresponding PN notations without semantic modification. In contrast to research that has been done by [13], they have been proposed formal semantic of BPMN that can be used to map to PN model. This paper tried to map all BPMN Notations, But the proposed mapping does not fully proper because identification of a number lacks in the BPMN standard specification. The proposed mapping missing with OR-join gateway, Exception handling for concurrent sub process instances and process instance completion. Therefore, [44] conducted research to solve the lack of examined result on PN after transformation and the multitude of BPMN elements, interpretation module of the raw verification result. They used the Model Driven engineering (MDE) approach to transforming and ATL rule to achieve transformation between BPMN to PN. Similarly, [12] proposed formal semantic for mapping BPMN to Colored PN (CPN). They claimed that mapping rule which has been proposed was able to fully BPMN model onto its corresponding CPN, but not for complex gateway issues, because doesn't have a default semantic.

YAWL-BPMN-YAWL. The transformation between BPMN to YAWL has very little attention. Straightforward transformation and the incompatibility concept of BPMN to YAWL were a reason [13]. BPMN and YAWL just propose several common concepts, such as task, gateway, and flow in have same concepts in YAWL. Reference [13] proposed tool BPMN2YAWL that solved the problem that YAWL does not accommodate the ability to handle chain several connectors together directly by proposing empty YAWL task. But the proposed method doesn't handle transaction and compensation handlers. Finally, the transformation doesn't handle complex gateways. In other, [42] [14] [43] proposed a formal semantic that can be used to map BPMN to YAWL. Proposed mapping rules provide on a core of BPMN notations based on syntactic rules, the transformation of ad-hoc sub-process with different attribute and BPMN message flow transformation. The paper further, they proposed exception handling, clarify a number of semantic lacks in BPMN specification and implemented their concept in open source plugin were developed by [13]. The result of transformation can be verified using YAWL-based verified tool after plugin in proM subsequently transform BPMN Models.

BPMN-UML AD. Similarly, PN to BPMN, bidirectional transformation between BPMN and UML AD have a little attention. There wasn't the publication of UML AD to BPMN. Reference [19] has transformed BPMN into UML-AD that realized by using a representation of both models in XML Metadata Interchange (XMI) as input and output. Transformation rules described in the form of an XSLT that meet the requirements on the tool independence and integration possibility, however, the weakness of this transformation process is the loss of information graphic layout of the model.

PN-EPC. The transformation from Petri Net to EPC have explored by [74], they present the algorithm for translating a one the class of PN, namely labeled P/T net, into an EPC and has been developed in the "Labeled WF-net to EPC" conversion plug-in of the ProM framework. This algorithm is done with a few steps, firstly, discover the join and split spheres for each labeled transition, produce each of these spheres a connector of the right type and produce an input event and a function, afterwards, the transition is transformed into an AND-join and AND-split combination, finally, produce the edges and clean the EPC notations by eliminating connectors with single input and output edges.

PN-UML AD. Unlike bidirectional between BPMN and UML AD, PN-UML AD has more attention, but not vice versa. Reference [67] has transformed UML AD to Petri Net based on three transformation rules i.e. the transformation rules for a state activity, transformation rules for the decision and transformation rules for state activity with more than one transition to leave. On the other hand, [68] have transformed based upon the execution semantics of state machines, they have acquired Message Sequence Charts (MSC) to visualize simulation results. Reference [64] have proposed translating rule, namely AD-CPNI rules, to translate every element of the activity diagram in accordance with the element Colored Petri Net which consists of eight steps: mapping variables, initial node, final node, action node, merge node, decision node, join node, and fork node. This approach assumes that the activity diagram well defined as well as all the parameters and input/output pins filled in correctly. Reference [65] proposed another approach to transforming or translate UML-AD to CPN. Because Petri Diagram Net more complex, have more nodes and edges, and is not suitable for presented to stakeholders, [65] introduces a solution for this problem, is to transform the UML activity Diagram (UML AD) into a Fundamental Modeling Concepts Petri Net Diagram (FMC-PND) compact notation, which can be converted to CPN for execution and validation.

UML AD-YAWL. Reference [65] have transformed UML-AD to YAWL with three-step method, firstly, object flows and nodes of UML-AD are translated to YAWL control flows by appending the important data to the appropriate task nodes, the next step, analyzed and classified the control-flow pattern that supports UML-AD based on their transformation either by pattern-to-pattern methods or element-to-element translate, and the last step, transformed the segment of model remaining with a way of element-to-element translate based on UML AD meta-models and the defined YAWL meta-models.

VI. CONCLUSIONS

The majority of the model transformation approaches is EPC to BPMN transformation. Commonly, the reason for this study is EPC is excellent integration in the ARIS Toolset and used very wide [26]. BPMN is one of the popular and interest in the industry. In other, PN to UML AD has more attention than other. BPMN to YAWL has the little attention of research community because of miss conception between BPMN and YAWL [13]. Authors have opinions that transformation between BPMN to UML is very important because transformation can solve problems of time, cost and quality to develop software in the scope of Model Driven Development, But still 1 study that discussed it.

There are 2 level of abstraction model transformation, higher-level abstraction to map rules between horizontal models and low level abstraction that execute to the target model. This research focused in the higher level, in general researcher used mapping transformation rule and add semantic techniques. There are little studies that proposed full mapping from the source model into target model, i.e. [26] and [13] proposed all BPMN notations can map fully to EPC or PN. But, the are still some lacks, in case [26] for handle message flow in BPMN, he used a semantic technique that hard code keyword "*sending*" and "*receiving*". In Other, [13] also identify the identity lacks BPMN standard. He missed OR-join gateway and concurrent sub process instance in their study.

Consequently, there are still many gaps and problems to be solved for future research in model transformation. In case model transformation survey, future work should be focused on low-level abstraction of model transformation.

REFERENCES

[1] V. Kotsev, I. Stanev, and K. Grigorova, "BPMN-EPC-BPMN Converter (PDF Download Available)," 2011. [Online]. Available: https://www.researchgate.net/publication/265401318_BPMN-EPC-BPMN_Converter. [Accessed: 01-Feb-2017].

[2] Nadja Damij and Talib Damij, "A Process-Oriented Modelling Technique," IEEE, 2009, pp. 134-139.

[3] M. Murzek and G. Kramler, "Business process model transformation issues," Proc. 9th Int. Conf. Enterp. Inf. Syst. Madeira, Port., vol. 3, pp. 144–151, 2007.

[4] K. Czarnecki and S. Helsen, "Classification of Model Transformation Approaches," 2nd OOPSLA'03 Work. Gener. Tech. Context MDA, pp. 1–17, 2003.

[5] G. Decker and W. Tscheschner, "Transformation from EPC to BPMN," in EPK 2009. 8. Workshop der Gesellschaft für Informatik e.V. (GI) und Treffen ihres Arbeitkreises "Geschäftsprozessmanagement mit Ereignisgesteuerten Prozessketten (WI-EPK). Gesellschaft für Informatik, 2009, pp. 91–109.

[6] V. Hoyer, E. Bucherer, and F. Schnabel, "Collaborative e-business process modelling: Transforming private EPC to public BPMN business process models," in Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 2008, vol. 4928 LNCS, pp. 185–196.

[7] O. Levina, "Measuring information content change in EPC to BPMN business process model transformation," in AMCIS 2012, 2012, vol. 4, p. 3032.

[8] I. Raedts, M. Petkovic, Y. Y. S. Usenko, J. M. E. M. van der Werf, J. F. Groote, and L. J. Somers, "Transformation of BPMN Models for Behaviour Analysis.," Msvveis, pp. 126–137, 2007.

[9] F. . Kog, R. J. . Scherer, and A. . Dikbas, "Petri Net based verification of BPMN represented configured construction processes," eWork Ebus. Archit. Eng. Constr. - Proc. Eur. Conf. Prod. Process Model. 2012, ECPPM 2012, pp. 243–249, 2012.

[10] R. M. Dijkman, M. Dumas, and C. Ouyang, "Formal semantics and analysis of BPMN process models using PN," Tech. Rep., vol. 50, no. 12, pp. 1–30, 2007.

[11] P. Kasar, "Business Process Verification using Formal Language Petri Net : An Approach," pp. 14–17, 2014.

[12] M. Ramadan, H. ; Elmongui;, and R. Hassan, "BPMN Formalisation using Coloured PN," Proc. 2nd GSTF Annu. Int. Conf. Softw. Eng. Appl., 2011.

[13] G. Decker, R. Dijkman, M. Dumas, and L. García-Bañuelos, "Transforming BPMN diagrams into YAWL nets," in Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 2008, vol. 5240 LNCS, pp. 386–389.

[14] Y. JianHong, S. ShiXin, L. Wen, and S. Wen, "Transformation of BPMN to YAWL," Proc. - Int. Conf. Comput. Sci. Softw. Eng. CSSE 2008, vol. 2, pp. 354–359, 2008.

[15] H. Bazoun, Y. Bouanan, G. Zacharewicz, Y. Ducq, and H. Boye, "Business process simulation: Transformation of BPMN 2.0 to DEVS models (WIP)," Simul. Ser., vol. 46, no. 4, pp. 141–147, 2014.

[16] S. Boukelkoul and R. Maamri, "Optimal Model Transformation ofBPMN to DEVS," 2015.

[17] D. Cetinkaya, A. Verbraeck, and M. Seck, "Model transformation from BPMN to DEVS in the MDD4MS framework," Proc. 2012 Symp. Theory Model. Simul. - DEVS Integr. M&S Symp., p. 28:1-28:6, 2012.

[18] Y. Rhazali, Y. Hadi, and A. Mouloudi, "Transformation Method CIM to PIM : From Business Processes Models Defined in BPMN to Use Case and Class Models Defined in UML," Int. J. Comput., vol. 8, no. 8, pp. 1467–1471, 2014.

[19] O. Macek and K. Richta, "The BPM to UML activity diagram transformation using XSLT," CEUR Workshop Proc., vol. 471, pp. 119–129, 2009.

[20] Y. Lemrabet, J. Touzi, D. Clin, M. Bigand, and J.-P. Bourey, "Mapping of BPMN models into UML models using SoaML profile," 8th Int. Conf. Model. Simul. - MOSIM'10, p. 6, 2010.

[21] M. Murzek and G. Kramler, "The top 7 adversaries encountered at defining model transformations ∗," 2006.

[22] J. Mendling, K. B. Lassen, and U. Zdun, "On the transformation of control flow between block-oriented and graph-oriented process modelling languages," Int. J. Bus. Process Integr. Manag., vol. 3, no. 2, p. 96, 2008.

[23] N. Lohmann, E. Verbeek, and R. Dijkman, "Petri net transformations for business processes–a survey," Trans. PN other, 2009.

[24] K. Grolinger, M. A. M. Capretz, A. Cunha, and S. Tazi, "Integration of BPM and Web services: A survey," Serv. Oriented Comput. Appl., vol. 8, no. 2, pp. 105–128, 2014.

[25] H. Kim, S. Kang, J. Baik, and I. Ko, "Test cases generation from UML activity diagrams," Softw. Eng. Artif., 2007.

[26] W. Tscheschner, "Transformation from EPC to BPMN," Bus. Process Technol., 2006.

[27] F. Kog and M. Gök, "Petri Net based VerificatioN aNd recoNfiguratioN of bPMN rePreseNted coNfigured coNstructioN Processes," Proc. 25th Conf. "Forum Bauinformatik", 2013.

[28] C. Ouyang, W. M. P. van der Aalst, W. Van Der Aalst, M. Dumas, and a H. M. ter Hofstede, "Translating bpmn to bpel," BPM Cent. Rep. BPM-06-02, BPMcenter. org, pp. 1–22, 2006.

[29] W. M. P. van der Aalst, "Business Process Management : A Comprehensive Survey," ISRN Softw. Eng., vol. 2013, pp. 1–37, 2013.

[30] L. Meertens, M. Iacob, and S. Eckartz, "Feasibility of EPC to BPEL model transformations based on ontology and patterns," Int. Conf., 2009.

[31] T. Mens and P. Van Gorp, "A taxonomy of model transformation," Electron. Notes Theor. Comput. Sci., 2006.

[32] K. Czarnecki and S. Helsen, "Classification of model transformation approaches," Tech. Context Model Driven …, 2003.

[33] M. Strommer and M. Wimmer, "A framework for model transformation by-example: Concepts and tool support," Objects, Components, Model. Patterns, 2008.

[34] M. Strommer, "Model transformation by-example," 2008.

[35] Tratt, Laurance, "Model Transformations and Tool Integration," Software & Systems Modeling, Vol. 4, No. 2, pp 112-122, 2005.

[36] T. Gardner, C. Griffin, and J. Koehler, "A review of OMG MOF 2.0 Query/Views/Transformations Submissions and Recommendations towards the final Standard," MetaModelling MDA, 2003.

[37] F. Jouault, F. Allilaire, J. Bézivin, and I. Kurtev, "ATL: A model transformation tool," Sci. Comput. Program., 2008.
[38] R. K. L. Ko, S. S. G. Lee, and E. Wah Lee, "Business process management (BPM) standards: a survey," Bus. Process Manag. J., vol. 15, no. 5, pp. 744–791, 2009.
[39] L. O. Johansson, M. Wärja, and S. Carlsson, "An evaluation of business process model techniques, using Moody's quality criterion for a good diagram," in CEUR Workshop Proceedings, 2012, vol. 963, pp. 54–64.
[40] J. Mendling and J. Ziemann, "Transformation of BPEL processes to EPCs," Proc. 4th GI Work. Event-Driven, 2005.
[41] L. Meertens, "EPC to BPEL transformations," 2009.
[42] H. Y. Jian, X. S. Shi, S. Wen, and J. W. Li, "Formal semantics of BPMN process models using YAWL," Proc. - 2008 2nd Int. Symp. Intell. Inf. Technol. Appl. IITA 2008, vol. 2, pp. 70–74, 2008.
[43] J. H. Ye and W. Song, "Transformation of BPMN diagrams to YAWL nets," J. Softw., vol. 5, no. 4, pp. 396–404, 2010.
[44] S. Mouline and A. Lyazidi, "Formal Verification of BPMN Models using PN," Maroc 2013, 1st Int. Work. Model. Algorithms Reliab. Open Comput., no. April, pp. 0–4, 2013.
[45] J. Recker and J. Mendling, "On the translation between BPMN and BPEL: Conceptual mismatch between process modeling languages," 18th Int. Conf., 2006.
[46] E. Biermann and C. Ermel, "Transforming BPMN to BPEL with EMF Tiger," Proc. Work. Graph-based, 2009.
[47] [M. Wei, Y. Xia, and J. Wei, "Model transformation from BPMN to BPEL2. 0," Appl. Res. Comput., 2008.
[48] Z. Haiyan, "Research on the Conversion Between BPMN and BPEL in the EERP," Comput. Telecommun., 2008.
[49] S. Gong and J. Xiong, "Interaction mismatch discovery based transformation from BPMN to BPEL," Serv. Comput. 2009. SCC'09. IEEE, 2009.
[50] Z. Shi, X. Zeng, T. Zhang, S. Huang, Z. Qi, and H. Li, "Bidirectional transformation between BPMN and BPEL with graph grammar," Comput. Electr., 2016.
[51] S. Mazanek and M. Hanus, "Constructing a bidirectional transformation between BPMN and BPEL with a functional logic programming language," J. Vis. Lang. Comput., 2011.
[52] R. Jubeh, "BPMN2BPEL transformation with Fujaba-a Case Study," 2009.
[53] O. Muliawan, B. Meyers, and D. Janssens, "BPMN2BPEL using MoTMoT," 2009.
[54] [54] M. De Mol and M. Zimakova, "A GROOVE Solution for the GraBaTs' 09 BPMN to BPEL Model Case Study," is.tm.tue.nl.
[55] M. de Mol and M. Zimakova, "A groove solution for the bpmn to bpel model transformation," 2009.
[56] R. Boussetoua, H. Bennoui, and A. Chaoui, "An automatic approach to transform BPMN models to Pi-Calculus," AICCSA), 2015 IEEE, 2015.
[57] M. Jurišić, "Transition between process models (BPMN) and service models (WS-BPEL and other standards): A systematic review," J. Inf. Organ. Sci., 2011.
[58] M. Indulska, J. Recker, P. Green, and M. Rosemann, "Are we there yet? seamless mapping of bpmn to bpel4ws," 2007.
[59] Z. Chen and Z. Zhongneng, "Model Boundary Defining Method of Mapping from BPMN to BPEL4WS," Comput. Eng., 2007.
[60] M. Jung, H. Kim, M. Jo, K. Tak, H. Cha, and J. Son, "Mapping from BPMN-Formed Business Processes to XPDL Business Processes.," ICEB, 2004.
[61] J. Park, M. Jung, M. Jo, and H. Kim, "BPMN2XPDL: Transformation from BPMN to XPDL for a business process," KIPS Trans., 2006.
[62] Z. Han, L. Zhang, J. Ling, and S. Huang, "Control-flow pattern based transformation from UML activity diagram to YAWL," Lect. Notes Bus. Inf. Process., vol. 122 LNBIP, no. July 2005, pp. 129–145, 2012.
[63] Z. Han, L. Zhang, and J. Ling, "Transformation of UML Activity Diagram to YAWL," Enterp. Interoperability IV, Mak. Internet Futur. Futur. Enterp., no. 2007, pp. 289–299, 2010.
[64] N. Maneerat, "Translation UML Activity Diagram into Colored Petri Net with Inscription," 2016.
[65] T. S. Staines, "Intuitive mapping of UML 2 activity diagrams into fundamental modeling concept Petri net diagrams and colored PN," Proc. - Fifteenth IEEE Int. Conf. Work. Eng. Comput. Syst. ECBS 2008, pp. 191–200, 2008.
[66] J. P. López-Grao, J. Merseguer, and J. Campos, "From UML activity diagrams to Stochastic PN," Proc. fourth Int. Work. Softw. Perform. - WOSP '04, p. 25, 2004.
[67] I. Trickovié, "Formalizing Activity Diagram of Uml By PN," Emisamsorg, vol. 30, no. 3. pp. 161–171, 2000.
[68] Z. Hu and S. M. Shatz, "Mapping UML Diagrams to a Petri Net Notation for System Simulation," Seke, pp. 213–219, 2004.
[69] Y. Hlaoui and L. Benayed, "A Model Transformation Approach Based on Homomorphic Mappings between UML Activity Diagrams and BPEL4WS Specifications of Grid Service Workflows," COMPSACW), 2011 IEEE 35th Annu., 2011.
[70] S. Iyengar, "Business process integration using UML and BPEL4WS," Components Futur. Softw. Eng., 2003.
[71] L. SHEN, F. SUI, F. LI, B. LEI, and L. BAI, "Workflow modeling with extended UML activity diagrams and its transformation into XPDL," Comput. Integr., 2009.
[72] B. Gallina, N. Guelfi, and A. Mammar, "Structuring business nested processes using UML 2.0 activity diagrams and translating into XPDL," XML Integr. …, 2005.
[73] S. Pornudomthap and W. Vatanawood, "Transforming YAWL workflow to BPEL skeleton," Softw. Eng., 2011.
[74] H. Verbeek and B. van Dongen, "Translating labelled P/T nets into EPCs for sake of communication," 2007.
[75] M. Weidlich, G. Decker, A. Großkopf, and M. Weske, "BPEL to BPMN: the myth of a straight-forward mapping," Conf. Move …, 2008.
[76] A. Brogi and R. Popescu, "From BPEL processes to YAWL workflows," Int. Work. Web Serv. Form., 2006.
[77] S. Hinz, K. Schmidt, and C. Stahl, "Transforming BPEL to PN," Int. Conf. Bus. Process, 2005.
[78] H. Zha, Y. Yang, J. Wang, and L. Wen, "Transforming XPDL to PN," Int. Conf. Bus., 2007.