Received November 26, 2017, accepted January 1, 2018, date of current version March 13, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2791666

Event-Driven Process Chain for Modeling and Verification of Business Requirements–A Systematic Literature Review

ANAM AMJAD[1], FAROOQUE AZAM[1], MUHAMMAD WASEEM ANWAR[1], WASI HAIDER BUTT[1], AND MUHAMMAD RASHID[2]

[1]Department of Computer Engineering, College of Electrical and Mechanical Engineering, National University of Sciences and Technology, Islamabad 44000, Pakistan

[2]Computer Engineering Department, College of Computer and Information Sys, Umm Al-Qura University, Mecca 21421, Saudi Arabia

Corresponding author: Muhammad Waseem Anwar (waseemanwar@ceme.nust.edu.pk)

ABSTRACT Automation of any business process primarily requires the identification of clear and precise requirements. However, the initially collected business requirements are usually expressed in natural language that creates ambiguities among different stakeholders. To overcome this problem, various business process modeling languages (BPMLs) have been introduced to represent the business requirements graphically. In this context, event-driven process chain (EPC) is a well-known BPML that supports the modeling and verification of business requirements in early automation phases. Although EPC is frequently researched to improve its modeling and verification capabilities, there is no study available yet to the best of our knowledge that examines and summarizes the latest EPC developments. Therefore, in this article, we comprehensively investigate the latest EPC approaches, trends, and tools for the modeling and verification of business requirements. Particularly, a systematic literature review is carried out to select and analyze 73 research studies published during 1998–2017. Consequently, the selected studies are classified into six categories, i.e., modeling (14), transformation (13), verification (17), general (20), semantics (5), and requirement (4). Moreover, latest EPC modeling approaches are identified and analyzed, i.e., UML (2), meta-model (3), integration (5), and EPC notations (4). Furthermore, EPC verification methods are also investigated, i.e., EPC (6), petri-nets (8), and other languages (3). Finally, 25 leading EPC tools have been presented, i.e., existing tools (14), proposed/developed tools, (5) and additional tools (6). It has been concluded that EPC provides adequate approaches and tool support for the modeling and verification of simple business requirements through atomic events. However, the complex business requirements cannot be modeled and verified through EPC due to the lack of complex event processing. Consequently, there is a strong need to include the support for the modeling and verification of complex events in EPC to manage multifaceted business requirements.

INDEX TERMS BPML, EPC, SLR, EPC verification, EPC tools.

I. INTRODUCTION

The clear and precise requirements are the foremost step in business process automation. However, the initially collected requirements are usually expressed in natural language. This leads to develop several ambiguities between different stakeholders as each has its own understanding about the textual requirements. Consequently, there is always a lack of consensus between stakeholders about the initially collected business requirements that severely affect the automation process in later stages. To overcome this problem, several Business Process Modeling Languages (BPML's) have been introduced to represent the business requirements graphically. Such representation significantly reduces the chances to misinterpret the business requirements, thus developing strong consensus among all stakeholders. Furthermore, BPML's also open the doors to verify the correctness of business requirements in early automation stages.

Several BPML's have been introduced so far to model the business requirements. Business Process Modeling Notation (BPMN) [1], Business Process Execution

Language (BPEL)[2], Object-Oriented Role Analysis Methodology (OORAM)[3], [4], Enterprise Distributed Object Computing (EDOC)[4], [5], Workflow Process Definition Language (WPDL) [4], [6], Electronic Business using XML (ebXML) [4], [7], Web Services Choreography Description Language (WS-CDL) [4], [8] and Business Process Definition Meta-model (BPDM) [4], [9] are the examples of well-known BPML's. These languages not only support the modeling of business requirements but also provide verification/analysis features. For example, BPMN is OMG (Object Management Group) standard that provides sophisticated modeling and verification features with sufficient tool support e.g. ARIS express, ADONIS and MagicDraw etc.

Businesses and institutions are in a constant phase to collect the data in order to offer or improve their existing processes [10]. In this context, Event-driven Process Chain (EPC) is one of the well-known Business Process Modeling Language (BPML). EPC resides in the category of traditional process modeling languages [4]. The goal to develop model can be described as a complete insight to the processes being modeled and the prediction of the behaviors which are not observed yet [11]. EPC language provides the modeling and verification capabilities to its users. EPC was developed by using the concepts of stochastic network and Petri nets within the framework of Architecture of Integrated Information System (ARIS) in the University of Saarland, Germany in 1992 by IDS Scheer [4]. EPC language has defined syntax and semantics. The syntax of EPC consists of events, function, logical operator and additional process objects. EPC always starts with an event called starting event. EPC ends on the event called ending event. Events are used to define the pre and post condition of the function. Events and functions have one incoming and one outgoing control flow arc. An operator is used to connect several events and functions. Logical operators are used for decision making. Additional process objects are optional in EPC and used to show the deliverable or information objects in EPC. The few important semantics of EPC [4] are as follows:

1. Events cannot make decisions like OR/XOR decisions. Events can only be linked with AND operator.
2. Functions can be associated with all three logical operators (AND, OR, XOR) for decision making.
3. Additional process objects can only be connected with the functions of EPC.

EPC is a graphical language consists of mainly three views i.e. Data view, Function view and Organization view. Data view consists of events and statuses [4]. Function view contains functions which represent the activities and the organization view helps in associating organizational role or unit with function. An additional control view connects these above three views with the help of connector [12]. EPC is known for its sophisticated modeling and verification features. It provides simple and easy to understand graphical notations for business process modeling. The major EPC activities are shown in **Fig.1**. The first step of EPC modeling

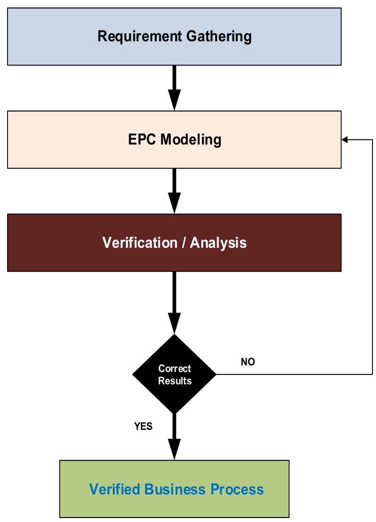

FIGURE 1. Major EPC activities.

and verification process is the collection of business requirements. In second step, the requirements are modeled through different EPC notations. Several tools are available (e.g. ARIS tool [13]) to support the modeling of EPC notations. In third step, EPC models representing business requirements are transformed into target verification/analysis models (e.g. Petri-nets). Subsequently, the verification/analysis is carried out to validate the correctness of business requirements. In case of errors, the appropriate changes have been made in the EPC model. Finally, the verified business process model is available for further implementation as shown in **Fig.1**. It is important to note that transformation step is not always required because few existing tools provide built-in support for verification/analysis of EPC models. One of its examples is EPCTools [14] which is an open source eclipse plugin having inbuilt modeling, simulation and analysis capabilities for EPC.

Although EPC is frequently researched to improve its modeling and verification capabilities, there is no study available yet to the best of our knowledge that examines and summaries the latest EPC developments. Therefore, in this article, we comprehensively investigate the latest EPC approaches, trends and tools for modeling and verification of business requirements. We try to find the answers of the following research questions:

RQ1: What are the leading approaches/techniques reported so far to improve the EPC modeling?

RQ2: What are the major transformation strategies to transform EPC models into other target models for further analysis/verification?

RQ3: What are the significant methods utilized for the verification of EPC models?

RQ4: What are the leading tools utilized/proposed for the modeling and verification of EPC?

RQ5: Is it possible to integrate EPC with other BPML's?

RQ6: Is it possible to model and verify complex/large business requirements through EPC?

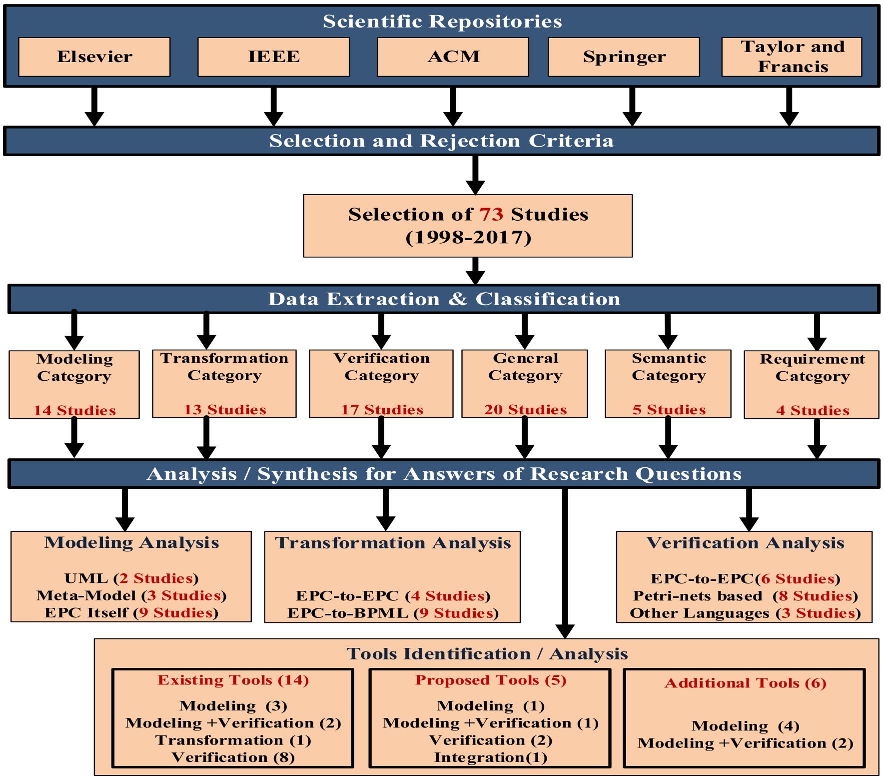

FIGURE 2. Overview of systematic literature review.

To answer the aforementioned RQ's, a Systematic Literature Review (SLR) is performed to select and examine the 73 research studies [14]– [84] published during 1998-2017. The overview of SLR is shown in **Fig.2**. The contributions of this paper are summarized as follows:

- Firstly, this study comprehensively investigates and summarizes the latest EPC developments for modeling and verification of business requirements. Particularly, 73 research studies published during 1998-2017 are considered. To the best of our knowledge, this is the first article where EPC is thoroughly analyzed and findings are summarized at one place.
- Secondly, this study identifies and analyzes 25 leading EPC tools for modeling and verification of business requirements. Such analysis certainly benefits the researchers/practitioners while selecting the right tool as per requirements.
- Finally, this study highlights the significant research gaps where improvements are required in EPC in order to model and verify the complex and large business requirements.

A review protocol is developed (**Section** II) to perform this SLR. Firstly, six categories are defined (**Section 2.1**) to simplify the data extraction and synthesis process. We consider five scientific repositories (i.e. IEEE, Springer, ACM, Elsevier and Taylor and Francis) for search process (**Section 2.3**) as defined in selection and rejection rules (**Section 2.2**). Consequently, we identify 73 research studies fully compliance with selection and rejection rules. We define the complete template (**Section 2.5**) to extract and analyze selected studies. As a result, selected studies are classified (**Section** III) into six categories as shown in **Fig.2**. Subsequently, all categories are comprehensively analyzed i.e. Modeling (**Section 3.1**), Transformation (**Section 3.2**), Verification (**Section 3.3**), General (**Section 3.4**), Semantic (**Section 3.5**) and Requirement (**Section 3.6**). Furthermore, several EPC tools are identified and analyzed in **Section 3.7**. The comprehensive analysis of selected studies leads to provide the answers of the RQ's (**Section 4**). The significant findings are discussed in **Section 5**. Finally, conclusion is given in **Section 6**.

II. REVIEW PROTOCOL

There are seven elements of review protocol. The two elements (i.e. Background and Research questions) are already given in Introduction (**Section** I). Therefore, we are omitting the details of these two elements in this section. The other five elements are described in subsequent sections.

A. CATEGORIES DEFINITION

We define six categories to simplify the data extraction and synthesis process. The description of each category is given below.

TABLE 1. Summary of search terms with results.

Sr.#	Search terms/ Keywords	Operator	IEEE	Springer	ACM	Elsevier	Taylor and Francis
1	EPC	N/A	16	146	75	548	206
2	Model Driven EPC	AND	0	0	4	0	0
		OR	890	940	1120	1045	205
3	EPC Extension	AND	0	6	18	1	1
		OR	2045	2597	3063	2890	870
4	EPC Transformation	AND	0	3	0	1	1
		OR	1378	2931	4385	3432	567
5	Verification of EPC	AND	0	7	0	5	0
		OR	944	1233	540	1643	79
6	EPC Requirement	AND	0	4		3	1
		OR	1016	876	1502	2248	876
7	EPC Semantics	AND	0	8	0	6	1
		OR	656	918	1287	1403	290

- **Modeling Category:** The research studies particularly dealing with the modeling of EPC are placed in this category. This includes the studies that are based on MDA (Model-Driven Architecture) e.g. the mapping of EPC to various UML diagrams, meta-models for EPC etc. In addition, the research studies that enhance EPC modeling notations are also placed under modeling category. Furthermore, modeling notations of EPC are also integrated with other languages to represent the requirements of complex and large systems. The research studies dealing with the integration of EPC are also placed under this category.
- **Verification Category:** The research studies dealing with the verification of EPC are placed in this category. It is important to note that it is essential to first develop the models of EPC before verification. However, in verification category, only those studies are considered where the verification is the primary focus and the details of EPC models are narrowly discussed.
- **Semantics Category**: As EPC is an informal language, its syntax and semantics are not well defined that lead to several ambiguities. Therefore, the research studies dealing with the improvement of syntax and semantics of EPC are placed under semantics category.
- **Transformation Category**: This category includes the research studies that focus transformation strategies to transform the given EPC models into other target models (e.g. Petri-nets, workflow models etc.) for further analysis/verification.
- **Requirement Category**: There are research studies that particularly deal with the requirement gathering and specification methods in the context of EPC. All such studies are placed under this category.
- **General Category:** There are research studies that can belong to the more than one aforementioned categories simultaneously e.g. research studies dealing with the modeling, transformation and verification of EPC altogether. Such research studies are placed under general category.

B. SELECTION AND REJECTION CRITERIA

We have defined logical selection and rejection rules to carry out this SLR in order to attain the desired objectives. The selection and rejection rules are summarized below:

- We only select the research studies that are highly relevant to EPC and must belong to one of our pre-defined categories (**Section 2.1).** We discard such research studies where EPC is partially discussed.
- We consider the research studies published during 1998 to 2017 in order to provide the complete insight of EPC developments.
- We select five reputed scientific repositories (i.e. IEEE, Springer, ACM, Elsevier and Taylor and Francis) to perform this SLR. Therefore, we only select the research studies that are published in one of the aforementioned repositories. Research studies from all other databases are not considered.
- The research studies having almost similar research contents are discarded and only one of them is selected.

C. SEARCH PROCESS

We initiate the search process by exploiting five repositories (i.e. IEEE, Springer, ACM, Elsevier, Taylor and Francis) as specified in selection and rejection rules (**Section 2.2**). We have utilized several keywords to perform search process. The summary of key words is given in **Table 1**. The search process is carried out with two kinds of operators i.e. AND, OR. The results collected from AND operator are not sufficient so OR operator is also used. In addition, we have also utilized several advance search options that are provided by considered repositories e.g. "where keyword contains", "year span" etc. The results returned through given keywords are very large in number which are difficult to scan completely. The primary reason is that EPC keyword belongs to different concepts. For example, in network domain, EPC stands for Evolved Packet Core. Therefore, we have refined the results by using advance search options e.g. specifying the subject as business and engineering in Taylor and Francis etc. Furthermore, we used complete "Event Driven

TABLE 2. Summary of selected studies w.r.t scientific repositories and publication type.

Database	Type	References	Total
IEEE	Journal	[17]	31
	Conference	[16][20][23][25][26][27][28][31][32][34][37][39][40][52][56][58][60][61][62][64][68][69][71][72][73][77][83][84][85][86]	
Springer	Conference	[14][15][18][21][24][29][35][36][41][43][44][45][46][48][50][51][53][55][59][63][67][70][79][80]	27
	Journal	[38] [65] [74]	
ACM	Journal	Nil	7
	Conference	[22] [30] [33] [81] [42][47] [78]	
Elsevier	Journal	[19] [49] [57] [66][75] [76] [82]	7
	Conference	Nil	
Taylor & Francis	Journal	[54]	1
	Conference	Nil	

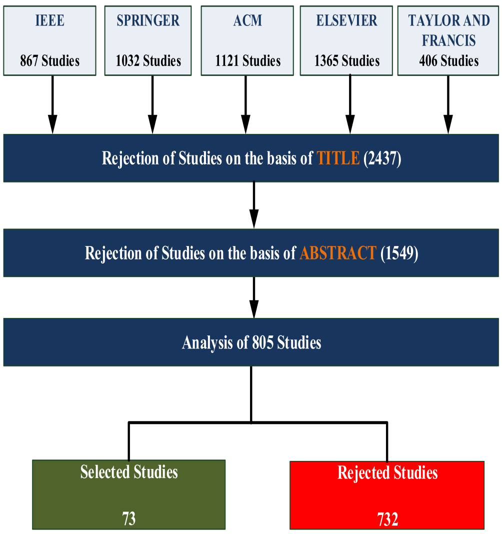

FIGURE 3. Summary of search process.

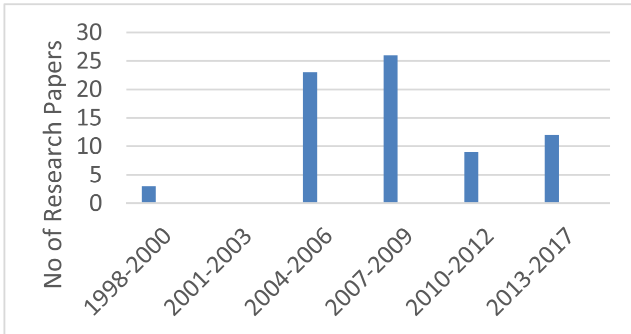

FIGURE 4. Distribution of selected studies w.r.t publication year.

Process Chain'' keyword instead of EPC to attain the desired results. Finally, we have selected 73 papers by following certain steps **Fig.3**:

- We overall consider 4791 studies and reject 2437 of them by reading Title.
- Subsequently, we consider remaining 2354 studies and reject 1549 studies from them by reading Abstract.
- We carefully examine the remaining 805 studies. On the basis of analysis, we reject 732 studies and select 73 studies that are completely conforming to our selection and rejection rules (**Section 2.2**).

D. QUALITY ASSESSMENT

We ensure the selection of high impact studies to guarantee the reliable outcomes of this SLR. For example, the selected databases are authentic and internationally accepted. We have selected 31 studies from IEEE, 27 studies from Springer, 7 studies from ACM, 7 studies from Elsevier and only 1 study from Taylor and Francis. The results given in **Table 2** demonstrate that we try to select high impact research studies as much as possible. Moreover, we also try to select studies that perform proper validation (e.g. experimental evaluation etc.) of the proposal. Furthermore, we also attempt to consider latest research studies as much as possible as shown in **Fig.4**. Consequently, we ensure the reliable and high impact outcomes of this SLR. In **Table 2**, the summary of scientific database with respect to the publication type is provided. **Database** represents the name of the database i.e. IEEE, Springer, ACM, Elsevier, Taylor and Francis. **Type** represents that whether the selected research paper belongs to the conference or journal. **References** are provided for each selected study. **Total** number of conference/ journal for every data base is given in the last column. It can be observed the large numbers of conferences are found in IEEE i.e. 30.

EPC language was invented in early nineties, therefore, we found first paper that is published in 1998. As we want to perform broader EPC investigation, we consider all papers published during 1998 to 2017 as shown in **Fig.4**. We found only three studies published during 1998-2001. There is no paper available on EPC between the ranges of 2001-2003 as shown in Fig.4. From 2004 to 2006, twenty-three studies are found in literature. The highest number of studies (twenty six) exists during 2007- 2009. From 2010-2012 and 2013-2017, nine studies and twelve studies are found respectively.

E. DATA EXTRACTION AND SYNTHESIS

We develop complete data extraction and synthesis template as given in **Table 3**. Firstly, we extract the bibliographic infor-

TABLE 3. Data extraction and synthesis template.

Sr. #	Description	Details
1	Bibliographic Information	Title, publication year and type of research paper (i.e. Conference or Journal) is analyzed.
2	Proposed methodology	Methodology followed by each selected study is observed.
3	Implementation details	Technologies used to implement the proposed methodology are observed.
4	Outcomes	Outcomes of each selected study are thoroughly analyzed
5	Grouping	Grouping of selected studies as per categories (**Section 3**). The results are summarized in **Table 4.**
6	Investigation of Categories	Analysis of each category to find the answers of the RQ's. The results are summarized as follows: Modeling Category (**Table 5, Table 6, Table 7** and **Table 8**), Transformation Category (**Table 9** and **Table 10**), Verification Category (**Table 11, Table 12** and **Table 13**), General Category (**Table 14**), Semantic Category (**Table 15**) and Requirement Category.
7	Tools	Tools used / proposed in the selected studies are analyzed(**Table 16** and **Table 17**)

TABLE 4. Grouping of selected studies into six categories.

Sr.#	Category	References of Corresponding Studies	Total
1	Modeling	[15][16][17][18][19][20][21][22][23][24][25][26][27][28]	14
2	Transformation	[29][30][31][32][33][34][35][36][37][38][39][40][41]	13
3	Verification	[42][43][44][45][46][47][48][49][50][51][52][53][54][55][56][57][58]	17
4	General	[14][59][60][61][62][63][64][65][66][67][68][69][70][71][72][73][74][75][76][77]	20
5	Semantics	[78][79][80][81][82]	5
6	Requirement	[83][84][85][86]	4

mation of each selected study. Secondly, we extract the core findings of each selected study like proposed methodology, implementation details and validation process. This provides the basis (data) to perform comprehensive analysis in order to achieve the objectives of SLR. Thirdly, we identified the tools that have been used or proposed by the selected studies. Finally, we perform detailed analysis to get RQ's answers as given in Introduction (**Section 1**).

III. RESULTS AND ANALYSIS

The selected studies are classified into six categories (**Section 2.1**) as given in **Table 4.** The references of corresponding studies are provided against each category for further exploration.

It can be seen from the **Table 4** that the modeling category comprises 14 studies. Basically, modeling is a core activity of EPC which is primarily required before performing the verification or transformation. The transformation category comprises 13 studies where manual or automated transformation is performed from one model to another model. There is a significant research on transformation of EPC in literature. Verification consists of 17 studies which shifts the

EPC semi-formal nature into formal by using formal mechanisms. In general category, 20 researches are found. Semantics and requirement category has 5 and 4 studies respectively. It can be observed that a lot of work is present regarding the modeling, verification and transformation which are main activities of EPC. However, efforts are made by researchers where EPC is also explored in other directions such as "requirements and EPC" as well. This categorization of studies (**Table 4**) simplifies the process of analysis/synthesis. To achieve the objective of SLR, we comprehensively investigate each category and present the precise outcomes. The details are given in subsequent sections.

A. MODELING CATEGORY

Modeling business requirements is foremost task of EPC. Therefore, researchers always try to improve the modeling capabilities of EPC. To achieve this, three approaches are commonly used i.e. Model Driven Architecture (MDA), enhancement of EPC notations and Integration of other elements with EPC modeling. MDA-based approach exploits novel model-driven trends to enhance the EPC modeling abilities. On the other hand, second approach deals with the addition of existing EPC notations to improve modeling. Third approach targets the enhancement of EPC models through integration. We divide the studies of modeling category into three aforementioned approaches to perform explicit analysis. The details are summarized in subsequent sections.

1) MDA-BASED MODELING APPROACH

MDA facilitates in developing the software systems with the help of models. Model represents an abstract level of the system being developed where Unified Modeling Language (UML) has contributed by introducing the UML diagrams. We identify two research papers where UML diagrams are exploited to enhance the modeling of EPC notations. We analyze the five significant UML diagrams (i.e. Activity, State machine, Use case, Component and Class diagram) that have been utilized in the concept of EPC modeling as

TABLE 5. Investigation of UML diagrams for EPC modeling.

Sr. #	Reference No	Activity diagram	Use case diagram	State machine diagram	Component diagram	Class diagram
1	[15]	✓	×	×	×	×
2	[16]	✓	✓	✓	✓	✓

TABLE 6. Summary of investigation for the extensions of EPC notations.

Sr. #	Reference No	Lack in EPC	Extension name	Tool support
1	[17]	Configurability	ADOM-EPC	N/A
2	[18]	Some workflow patterns	Yet another EPC	N/A
3	[19]	Configurability	C-EPC	N/A
4	[20]	Continuity processes	WDEPC	N/A

shown in **Table 5**. *Activity diagram* shows the sequence of the behavior of system. In use case diagrams, set of actions are performed by some external or internal actor. *State machine diagram* express the behavior of the system with the help of finite state machine concept. *Component diagram* shows the structure of the system being implemented in the form of components. *Class diagram* represents the structure of the system through various elements such as classes, features, dependencies and generalization.

Korherr and List [15] mapped EPC notations to the UML activity diagrams and developed a UML2 profile for EPC. A profile can extend the meta-model or another profile for specific application and it consists of stereotypes, constraints and tagged values. The profile being developed was tested on example of insurance claim process to represent the applicability of proposed methodology. Similarly, Loos and Thomas [16] mapped the EPC diagram into different UML diagrams like activity diagram, use case diagram, state machine diagram, component diagram and class diagram. The objective of this study is to change the process concepts into object orient concepts to reduce the gap between the business user and software developer.

2) EXTENSION-BASED MODELING APPROACH

EPC provides various notations to model business requirements. However, these notations are inadequate to model large and complex business requirements. Therefore, researchers frequently propose novel approaches to enhance the capabilities of EPC notations. There are two common strategies to extend EPC notions: 1) Simply Extend EPC notations 2) Enrich the meta-model of EPC. We found different studies pertaining to both categories as discussed in subsequent sections.

a: SIMPLE EXTENSION OF EPC NOTATIONS

EPC language has high user acceptance and wide tool support. Some extensions are proposed for EPC to make it suitable for specific application or environment. We investigate such EPC extensions in **Table 6** with following evaluation parameters: 1) **Lack in EPC** indicates the reason why this extension is proposed 2) **Extension Name** describes the name of the proposed extension 3) **Tool Support** evaluates whether the tool is developed or used to support the proposed extension.

In **Table 6**, the lack of certain elements in EPC provides the basis of the new extension. Lack of configurability [17], [19] means enterprise systems need to be configured to fit organizational requirements and another benefit is to provide support for business operations. On the other hand, there are many workflow patterns available. On the basis of these patterns, different process modeling languages are analyzed. Various studies have been carried out in order to increase the workflow pattern support for any BPML. This also happens for EPC, where lack of workflow pattern support is increased through extension. Also, lack of continuity processes involves risk, security and compliance which need to be modeled and reconfigured through EPC by its extension. Different extensions of EPC are proposed to overcome the lack in EPC.

Reinhartz-Berger *et al.* [17] propose new extension of EPC for reference models named Application-based Domain Modeling (ADOM)-EPC. It provides more reuse and adaptability for reference models by supporting configurability. Mendling *et al.* [18] propose a novel EPC extension named Yet another EPC (yEPC) to handle more workflow patterns: empty connector, multiple instantiation and cancelling concepts. Rosemann and Van der Aalst [19] present new extension for the configuration support called C-EPC for reference models. Brand *et al.* [20] introduce Workflow engine based and Dataflow Oriented EPC (WDEPC) and validate it through loan granting process. WDEPC consists of five views; first contains data storage, second view comprises of data store nodes, third and fourth consists of functions and view respectively and the last view has organizational aspect. The reason to extend EPC using WDEPC is to support the continuity process which was not possible to model before. It is analyzed that these four EPC extensions in **Table 6** doesn't have any tool support. Due to missing tool support, a wide applicability of these extensions is questionable.

TABLE 7. Summary of investigation for meta-model based EPC extensions.

Sr. #	Reference No	No. of extended elements	Modeling notations proposed	Tool support
1	[21]	7	Yes	N/A
2	[22]	4	Yes	N/A
3	[23]	2	Yes	N/A

b: EPC META-MODEL BASED EXTENSIONS

We found three papers where the extension of EPC is proposed in terms of meta-model extension as shown in **Table 7**. In this category, the meta-model of EPC is first extended and then new notations are proposed as per extended meta-model. We consider three evaluation parameters to analyze meta-models based EPC extensions as follows: 1) **No. of extended elements** describes the number of elements which are extended in EPC meta-model. 2) **Modeling notations proposed** indicates the presence of graphical notations for extension. 3) **Tool support** highlights the presence of tool for the proposed extension based on EPC meta-model.

Krumeich *et al.* [21] discuss various complex event patterns in their study. Among them, eight complex event patterns (event start time, event end time, event cardinality, event exclusion, event sequence, event location, event trend and data dependency) are extended using EPC meta-model to support these patterns. Graphical modeling notations are also proposed but there is no tool support available yet. Korherr and List [22] extended EPC meta-model to support the performance measure. Modeling notations of four extended elements (quality, cost, cycle time and measure flow connector) are proposed without any tool support. Similarly, Stefanov and List [23] extend only two elements through EPC meta-model (PM information object and PM flow connector) with additional process objects of EPC.

3) INTEGRATION-BASED MODELING APPROACH

EPC is an expressive graphical language which is used for modeling. It can be modeled using MDA and its modeling notations can be extended through simple or meta-model based extensions. Likewise, another significant area of EPC is its integration with other languages to make it suitable for particular environment. We analyze the integration possibilities of EPC in **Table 8**. The **proposed integration** represents the two languages which are integrated. **Task of 1st language** and **Task of 2nd languages** is used to represents the task associated e.g. modeling etc. **Applicability** indicates the application of the proposed integration in particular domains. Whereas, **Tool used** evaluates the tool (if any) which is used to perform the integration. Total five studies are found in this category.

van der Aalst *et al.* [24] integrate two EPC models into single process model. Both integrated EPC models are used for modeling of processes. The proposed integration is useful for process optimization of the companies offering purchasing, invoicing and HR processes. The authors apply ''function-graph'' algorithm and implementation is done through ProM plugin. In another study, Yu *et al.* [25] integrate two modeling languages; EPC and Modularized Event Driven Colored Petri–Net (MED-CPN). The reason behind this integration is that single language is not suitable for agile manufacturing cell where a lot of activities from modeling to operational need are necessary to be performed. Authors utilized Colored Petri Net (CPN) tool to demonstrate the applicability of proposal. Yu *et al.* [26] integrates EPC and Parametric Event-driven Flowed Token Petri Net (PEFT-PN) with the help of PEFT-PN tool.

In [24] and [25], first modules perform modeling at upper level and second module provide operational support at low-level. Another contribution is made by Dollmann *et al.* [27] by utilizing CoMoMod tool where modeling of EPC and Petri net is supported with EPC Markup Language (EPML). This integration is applicable in inter-organizational Business Process Management (BPM) where collaboration is important to track. Another interesting work in this area is performed by Liu and Wang [28] to integrate the EPC and complex event processing for the Radio Frequency Identification (RFID) middleware. EPC is used for modeling and complex events are extracted from RFID middleware. The BP modeling tool is used for integration.

From **Table 8**, it is analyzed that EPC is used as one of the component for integration. One common thing is the task associated with EPC language is to perform the modeling. The other language provides the operational support, modeling or extraction. These integrations are proposed for specific environments such as manufacturing cell, generic agile system, inter-organizational BPM or RFID business application. The advantage obtain by these integrations is a use of unified system rather than using two separate modules for different tasks.

B. TRANSFORMATION CATEGORY

Transformation provides the basis for the analysis/ verification of EPC models. In this category, transformation of EPC to its extensions and transformation of EPC to other business process modeling languages is presented. In **Table 9**, the transformation of EPC to its extensions are analyzed with the following parameters. 1) **Source language** is the input language which can be EPC or its extension. 2) **Target language** represents the name of language obtained as a result of transformation. 3) **Tool used** represents the tool which is

TABLE 8. Investigation of EPC integration with other languages.

Sr. #	Reference No	Proposed integration	Task of 1st language	Task of 2nd language	Applicability	Tool used
1	[24]	Merging two EPC models	Modeling	Modeling	Purchasing, invoicing	ProM
2	[25]	EPC and MED-CPN	Modeling	operational support	Manufacturing Cell	CPN
3	[26]	EPC and PEFT-PN	Modeling	operational support	Generic Agile System	PEFT-PN
4	[27]	EPC and PN	Modeling	Modeling	Inter organizational BPM	CoMoMod
5	[28]	EPC and Complex event processing	Modeling	Extraction of complex events from RFID	RFID Business applications	BP Modeling tool

TABLE 9. SUMMARY of EPC-to-EPC transformation strategies.

Sr. #	Reference No	Source Language	Target Language	Tool used	Algorithm used	Interchange format
1	[29]	C-EPC	Lawful EPC	N/A	N/A	EPML
2	[30]	C-EPC	Correct EPC	N/A	Derivation Algorithm	EPML
3	[31]	BPMN	EPC	ADONIS	N/A	N/A
4	[32]	Enterprise Models	eEPC	N/A	N/A	XPDL

TABLE 10. EPC transformation to other BPML's.

Sr. #	Reference No	Source language	Target language	Tool used	Algorithm used	Inter change format
1	[33]	yEPC	YAWL	N/A	Transformation algorithm	N/A
2	[34]	EPC	View process	View Process Demonstrator	N/A	EPML
3	[35]	EPC	BPMN	N/A	N/A	N/A
4	[36]	EPC	UML Activity Diagram	ATL transformation	N/A	N/A
5	[37]	EPC	BPMN	N/A	Shannon Information theory	N/A
6	[38]	EPC, Workflow net	GXL	N/A	N/A	N/A
7	[39]	EPC	BPMN	N/A	N/A	N/A
8	[40]	EPC	PMC	N/A	N/A	N/A
9	[41]	EPC	BPEL	Oracle BPA	N/A	N/A

used to perform transformation. 4) **Algorithm used** indicates the availability of transformation algorithm in the given study. 5) **Interchange format** indicates the language which is used for the transformation as an intermediate step. It is important to note that aforementioned five parameters are also used for evaluating transformation of EPC to other business process modeling languages as given in **Table 10**.

In **Table 9**, Recker *et al.* [29] transforms the Configured EPC (C-EPC) to the regular EPC. EPC Markup Language (EPML) is used for transformation. Mendling *et al.* [30] transform the Configured EPC to correct EPC in order to overcome the configuration problems. Configured EPC is an extension of EPC which is selected as source language. Derivation algorithm is used as a transformation algorithm and EPML are used as interchange format. Murzek *et al.* [31] transform the BPMN into EPC with ADONIS tool. This paper highlights the issues based on meta-model transformation and transformation is performed on structural patterns. In another approach, Weiqing *et al.* [32] transform the enterprise models into workflow model (eEPC) by mapping the function view, organization view, resource view of enterprise model into business modeling language i.e. EPC using XML Process Definition Language (XPDL).

From **Table 9**, it is analyzed that EPC to EPC transformation is utilized to make EPC language self-capable of verification or further analysis. For example, in studies [29] and [30], configured EPC is enriched by transforming it into lawful or correct EPC which can be verifiable in future. Also, the interchange formats used such as EPML or XPDL provide the basis for tool support. Although there is no tool support given in studies [29] and [30], tools can be developed in future through provided interchange format. There is only one study [30] that has used derivation algorithm to check the correctness and termination/deadlock of the system. Another study [31] has used ADONIS tool capable of transforming one language to EPC. The transformation of EPC to other BPML is analyzed in **Table 10**. Here, first study (Mendalinget *et al.* [33]) describes that YAWL is powerful workflow language which provides stronger support for workflow patterns. EPC (yEPC) extension can be transformed into YAWL with the help of transformation algorithm. In another study, Ziemann *et al.* [34] transform the EPC into view process or public process with the help of VPD tool through EPML interchange format which is the integration of EPC and XML. BPMN is a standardized BPML, therefore, Hoyer *et al.* [35] transform the private EPC language to the public BPMN language. Similarly, Levina [37] transforms the EPC to BPMN and information content is calculated by Shannon information theory. The finding of the paper claims that information loss by transforming EPC to BPMN is not significant. Subsequently, the mapping of private EPC to public EPC and then Public BPMN is presented. Likewise, Strommer *et al.* [36] transform the EPC into UML activity diagram at model level. At the meta-model level, ATL (Atlas Transformation Language) is used for transformation.

Winter and Simon [38] exchange the two graphical modeling languages (EPC and workflow nets) into XML based graph exchange language. Thomas and Leyking [39] mapped the concepts of EPC to BPMN and then to BPEL. The purpose of this transformation is to convert information model in EPC to conceptual modeling using BPMN. To support execution, BPMN is further transformed into BPEL. This process is useful for service oriented architecture. Meerteens *et al.* [41] map the EPC into BPEL by using Oracle BPA tool. Both languages are different as EPC is graph based and BPEL is block based. The proposed transformation considers the ontology and patterns for both languages. The execution of EPC models can be obtained by this transformation which helps in identifying errors.

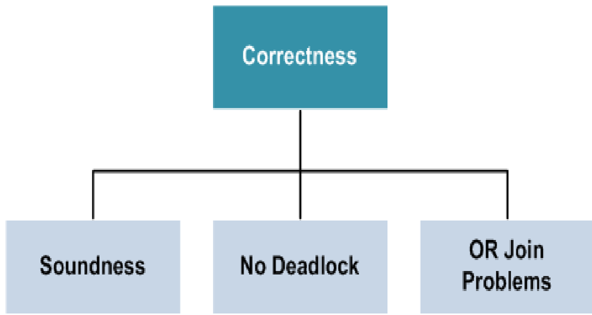

FIGURE 5. Important Verification Criteria's for EPC models.

From **Table 9** and **Table 10**, we have analyzed that EPC is transformed to another EPC extension or it is transformed to other BPML. However, these transformations are not practically more useful due to inadequate tool support. Few studies have contributed in introducing EPML as interchange format which forms the basis for the tool support. These studies have focused on the partial implementation which demands more work in this direction for practical achievements.

C. VERIFICATION CATEGORY

This category comprises the studies that are dealing with the early verification of EPC models. This ensures that the system need to be developed is free from many ambiguities or misinterpretations. The verification of EPC models usually perform through ***Correctness*** criteria [45]. The correctness can be further divided into three categories; soundness, No deadlock and OR-join as shown in **Fig. 5**.

- **Soundness:** Soundness represents one of the correctness criteria for the business process models. It has one source and one sink which offer the firing sequence from source to the sink.
- **No Deadlock:** It means system should terminate properly. Deadlock in the system halts the termination of the process.
- **OR-Join Problems:** OR join offers non local semantics where it has to wait for one, two or up to all incoming tokens. The presence of OR-join can be the reason of errors in EPC models.

In order to check the correctness of EPC, formal languages are used. It is also important to highlight that Petri net offering the set of graphical notations is widely/commonly used to verify the correctness of EPC. A Petri net is composed of places, transitions and arcs. A place is represented by circle. It is used to contain the discrete number of tokens. A transition is represented by the rectangle. It is used to fire tokens when it is enabled. The firing of transition consume token. The transition is called non-deterministic i.e., among many enabled transitions, anyone can be fired. In other words, Petri nets have well defined mathematical theory to verify the business processes. In this section, many algorithms such as reduction rules, reachability graph and state space utilize the Petri net theories [45].

TABLE 11. Verification of EPC in terms of EPC.

Sr. #	Reference No	Tools used	Algorithms used	Verification criteria
1	[42]	Microsoft Visio	Coloring algorithm	Correctness in terms of syntax
2	[43]	ProM	NIL	Correctness
3	[44]	ProM	Reduction rule	OR Join
4	[45]	ProM	Reduction rule	Soundness
5	[46]	ARIS, ProM, xoEPC	Reduction rule, Reachability graph	Deadlock
6	[47]	ARIS, ProM, xoEPC	Reduction rule, Reachability graph	Deadlock

TABLE 12. EPC to Petri-net based verification.

Sr. #	Reference No	Tool used	Algorithm used	Verification criteria
1	[48]	Net proof	Reduction algorithm, Petri-net theory	Correctness
2	[49]	ARIS, SAP R/3	Inspection, cover ability graph	Soundness, Deadlock
3	[50]	ProM 4.0	Heuristic Net	Correctness
4	[51]	EPCTools, ProM, YAWL	Fixed point semantic, Mendaling semantic on state and context, YAWL semantics	Soundness
5	[52]	ProM	Reachability graph	OR join
6	[53]	ProM	State space	OR join, soundness, deadlock
7	[54]	ProM	Reachability graph, theory of region	OR join
8	[55]	ProM	Reduction rule	Soundness

We investigate the verification approaches for EPC models through following parameters: 1) **Tool used** describes the name of the tool which is used for verification 2) **Algorithm used** indicates the algorithm selected for verification 3)**Verification criteria** refers to the methods used to check the correctness of EPC models as shown in **Fig. 5**.

In **Table 11**, only those studies are analyzed in which verification of EPC in terms of EPC language is performed. In first study, Smuts *et al.* [42] verify EPC models using a plug-in which is developed in Visio tool. Particularly, EPC rules are defined to detect major and minor errors using textual and graphical form. The coloring algorithm is used to identify the graphical errors. The focus of the results is the checking of effectiveness, efficiency, awareness and usefulness issue. Another interesting work is B.F. van Dongen *et al.* [43] that utilized ProM tool for EPC verification. Similarly, two more studies [44], [45] check the soundness of the EPC models through ProM. Particularly, the major contributions of these studies [44], [45] is to transform the informal semantics into formal semantics for EPC verification. The difference in both papers is in terms of verification criteria where one paper has focused on the OR join while other paper has checked the soundness of the models. Likewise, the studies [46], [47] calculate metrics and occurrence of errors. Important footstep of both studies [46], [47] is the correlation analysis between the error probability and metrics. Three tools are used for this purpose are ARIS, ProM and xoEPC. ARIS tool is used for modeling and XoEPC is used to generate the information on errors while ProM tool is used to analyze errors.

In **Table 12**, EPC language is selected as input language and it is converted to Petri nets for verification. For example, Langner *et al.* [48] apply Petri net theory to translate the EPC into Boolean net which is a sub class of Petri net. For translation, NetProof tool is developed. Similarly, van der Aalst [49] translates the concepts of EPC into Petri nets and soundness property of the EPC is verified. In [50], EPC diagram is imported into ProM tool where it offers analysis, conversion and verification of EPC models. Gruhn and Laue [51] analyze 712 models on three tools (EPCTools, ProM and YAWL) and their soundness properties are checked. Results show that these three tools are efficient to check the

TABLE 13. Other languages based verification of EPC.

Sr. #	Reference No	Conversions to other languages	Tool used	Algorithm used	Verification criteria
1	[56]	Casual footprint	ProM	Casual Graph	Soundness
2	[57]	YAWL	YAWL	wofYAWL	Relaxed soundness
3	[58]	Timed Automata, Time Petri net	UPPAAL, TAPAAL	N/A	Deadlock, reachability

soundness of the EPC models but some bugs are also identified and fixed.

Similarly, Mendling *et al.* [52] transform the EPC into Petri net. This approach utilized ProM tool and plug-in to convert the EPC into reachability graph then Petri net is generated and verified. The errors caused by OR join are highlighted and solution is proposed by conversion to Petri nets. Moreover, Lohmann *et al.* [53] transform the BPMN, EPC, BPEL and YAWL to Petri nets. Their transformation challenges and applications are explained. In context of EPC, transformation challenges occur for OR, XOR connector and multiple start and end events. Deadlock, OR join and soundness checking is the major contribution for verification. Additionally, van Dongen *et al.* [55] define the two-step approach for verification of EPC models. Firstly, EPC is translated into reduced EPC by using reduction rules in ProM tool. Secondly, the soundness of EPC models is verified by utilizing reduced EPC. The results show that out of 600 process models 34 are erroneous. Similarly, Amjad *et al.* [58] take the EPC as input. The mapping rules are provided which are used to map EPC into the timed automata and time Petri net. The tools used for timed automata and time Petri nets are UPPAAL and TAPAAL respectively. Verification of two properties; deadlock and reachability is presented. From **Table 13**, it is analyzed that EPC models can be converted into other languages to perform verification. This allows the existing tools of other languages (e.g. UPPAAL for timed automata etc.) to be used for the verification of EPC models. However, it is very important to keep the proposed conversion or mapping semantically equal.

Both EPC and Petri net languages belong to the business process modeling languages but conversion to Petri nets is formalized and offer many analysis techniques. From **Table 12**, it is analyzed that in order to achieve EPC to Petri net based verification, the ProM is the leading tool and commonly used algorithm is reduction rules. Soundness of the models is checked by four papers. Only one paper [49] has checked deadlock along with soundness. Also, [53] focused on eliminating problems with OR join.

In **Table 13**, we analyze the studies where EPC is converted into other languages. For example, Yet Another Workflow Language (YAWL), casual footprint, timed automata and time Petri net. Conversion /Mapping of EPC to these languages is the first step, then by using the equal semantics of target language, the verification is performed

In **Table 13**, ***Conversion to other languages*** is used to represent the target language selected for verification. We found three studies where EPC is transform to other languages for verification. For example, van Dongen *et al.* [56] verify soundness property of Petri net and EPC using casual footprint technique. Three kinds of event pattern which are deadlock, multiple terminations and trap are checked for soundness of EPC models. In another study, Mendling *et al.* [57] converts the EPC diagram into YAWL language and then verification is performed through YAWL tool.

D. GENERAL CATEGORY

Twenty (20) research papers are placed under this category. These research papers don't focus on one specific category e.g. modeling, verification etc. These research papers have provided an extensive work on more than one category of EPC. General category is created to avoid the confusion of placing the same paper in two or more than two different categories separately. We place all such studies in this category and analyze each study with respect to four major categories i.e. modeling (**Section 3.1**), transformation (**Section 3.2**), verification (**Section 3.3**) and semantics (**Section 3.5**). To demonstrate the importance of this category, **Table 14** is developed. This table is used to classify those research papers in which more than one categories i.e. modeling, verification, semantics or transformation are targeted simultaneously. **Table 14** demonstrates the targeted categories in each research paper with the help of respective tick and cross. The summary of research studies in general categories is provided in subsequent paragraphs.

Cuntz and Kindler [14] proposes the simulation of EPC models with EPC tools. Verification is done with the help of fixed point theorem and symbolic model checking. In addition, the syntax and semantics of EPC are also discussed. Similarly, van Hee *et al.* [59] extend EPC (eEPC - extended EPC) and transform it into colored Petri nets for verification. Another work (Zia *et al.* [60]) focuses on flexibility of system where changing requirements will not change the design of the EPC diagram due to flexibility achieved. In this paper, EPC models are created and transformed to Pi-calculus. Dong *et al.* [61] present the approach to extend the EPC model which can support web services. Integration of process layer and web service layer is performed and then

S-EPC (Semantics EPC) is mapped to the BPEL (Business Process Execution Language).

Kapuruge *et al.* [62] introduce a new textual language for EPC named as EPClets. EPML is a textual language already exists but EPClets is more expressive than EPML because it uses declarative event action rules to express control flows. The verification is accomplished through EPClets tool which is eclipsed based plugin developed in Java technology. In another interesting work, Mendling [63] defines the syntax, semantics, extension (Configurability) for EPC to perform verification in ProM tool. Finally, workflow patterns are compared for different languages; workflow net, UML activity Diagram, BPMN, EPC and YAWL. In another study, Pfeiffer *et al.* [64] apply the model checking technique for verification of workflow systems. The EPC models are developed in ARIS tool and pipe-line verification is implemented in Bendara tool. This approach is highly suitable for e-commerce systems. Another work (Wil M.P. van der Aalst *et al.* [65]) focuses on the correctness of the configurations of business modeling languages. EPC extension (Configured EPC) is verified in terms of soundness of the model. Kindler [66] defines the syntax and semantics of EPC to perform analysis, simulation and verification. Decker and Mendling [67] proposes the instantiation semantics for six business process modeling languages; Petri nets, UML activity Diagram, BPMN, EPC, YAWL and BPEL. These languages start with multiple start events leading to the misinterpretation. To overcome this problem, semantics of each language is defined with CASU framework. This framework is also used to identify errors in EPC models.

A new EPC extension (LightEPC) is introduced by Sidaxue *et al.* [68] to define resource loop patterns and data unreachability patterns for errors detection. Similarly, Mei *et al.* [69] introduce rule based EPC which extends the functionality of EPC with multi view support, business rule, maintain the semantics and execution of the language. This framework facilitates the service oriented architecture. In a slightly different way, Bogal *et al.* [70] develop process goal constructs from EPC. This transformation takes place with G-Tree construction algorithm. Feja and Fotsch [71] propose a model checking technique. Particularly, transformation of EPC to SVM models is performed. Finally, CTL (Computation Tree Logic) is used to perform verification. ARIS is used for modeling of EPC and AML (Aris Markup Language) is used for transformation. In another study, Irfan [72] model the business process of Supply Chain Management (SCM) using the EPC language. Tsai *et al.* [73] perform transformation of EPC to Workflow Initiative Formal Approach (WIFA). As EPC is informal language and does not provide the verification of modeled process, this conversion helps in verifying the well-formedness of the models created in EPC. Mendling and Nuttgens [74] introduce new XML based interchange format called EPML. It helps in elaborating the EPC syntax and its correctness.

Modeling and transformation is carried out by Sivri *et al.* [75] to present the process-oriented knowledge for automotive industry. For this purpose, EPC is used for modeling of the default setting of the system. In addition, EPC is carried out with two-step transformation. Firstly, EPC is transformed to the BPMN and then BPEL is finally generated. Another interesting work (Karnyongsiriwi *et al.* [76]) introduces Business Process as a Service (BPaaS) which is cloud computing technology. This framework has used two business process modeling languages; EPC and BPMN. Modeling and integration of these two languages is performed in this paper. The benefits of this approach are high level abstraction of BPaaS and extraction of configurable guidelines which are useful in the repository of SAP reference models. Schunselaar *et al.* [77] utilize the systematic evaluation technique for the development of EPC models for complete ERP system.

From **Table 14**, it is analyzed that different studies perform more than one EPC activities such as modeling, transformation or verification. Out of all 20 studies, 15 studies presented modeling. 13 studies have performed verification and 11 studies discussed the semantics of EPC while 7 studies performed transformation. The different combination of categories can also be observed from **Table 14**. For example, modeling and verification is concurrently performed by 8 studies. Modeling and semantic are simultaneously discussed in 7 studies. Modeling and transformation is performed in total six studies. Finally, EPC modeling, verification, semantic and transformation is performed altogether in only one study [59].

E. SEMANTIC CATEGORY

In this category, the selected papers dealing with syntax and semantics of EPC are placed. In **Table 15**, different parameters are used which are as follow: 1) **Syntax of EPC** represents the notations (events, functions, logical operators and additional process objects) used for modeling of EPC as discussed in the studies 2) **Semantics of EPC** represents the logical reasoning to construct/model the EPC 3) **OR Join** represents the studies targeting the OR join problem 4) **Comparison of EPC language with other languages** represents whether the selected studies compare the syntax and semantics of EPC with other languages.

In first study (**Table 15**), Santos *et al.* [78] discuss the EPC syntax and semantics on the basis of selected Unified Foundational Ontology (UFO). The advantage of this research is to clarify the concepts of EPC and removing the ambiguities to understand business modeling language. In another study, van Dongen *et al.* [79] use EPC as checking the similarity between business process models. In order to check the similarity between casual footprint and EPC, syntax and semantics of both languages are discussed in detail. The ProM tool is used for validation. Sarshar and Loos [80] compare the control flow of EPC and Petri net language for the end user. However, the syntax and semantic of EPC language is not presented but the problem of OR join is considered.

TABLE 14. Relationship of general category with other categories.

Sr. #	Reference No	Modeling	Verification	Semantic	Transformation
1	[14]	×	✓	✓	×
2	[59]	✓	✓	✓	✓
3	[60]	✓	✓	×	✓
4	[61]	✓	×	×	✓
5	[62]	✓	✓	✓	×
6	[63]	✓	✓	✓	×
7	[64]	✓	✓	×	×
8	[65]	✓	✓	×	×
9	[66]	×	✓	✓	×
10	[67]	×	✓	✓	×
11	[68]	✓	✓	✓	×
12	[69]	✓	✓	×	✓
13	[70]	✓	×	✓	✓
14	[71]	✓	✓	×	×
15	[72]	×	✓	×	✓
16	[73]	✓	×	✓	×
17	[74]	×	✓	✓	×
18	[75]	✓	×	×	✓
19	[76]	✓	×	✓	×
20	[77]	✓	×	×	✓

TABLE 15. Summary of studies dealing with EPC semantics.

Sr. #	Reference No	Syntax of EPC	Semantics of EPC	Identified OR operator problem	Comparison with other language
1	[78]	✓	✓	×	×
2	[79]	✓	✓	×	✓
3	[80]	×	×	✓	✓
4	[81]	×	✓	×	✓
5	[82]	✓	×	×	✓

In addition, Grossmann *et al.* [81] select five high level languages i.e. Workflow Nets, UML 2.0 activities, YAWL, BPMN and EPC. Some rules of inter process dependencies are defined and each language is checked against it. UML 2.0 extension is proposed to model four kinds of dependencies. Similarly, Jost *et al.* [82] provides an empirical investigation to compare three languages (i.e. BPMN, EPC and UML activity diagram) in the context of understandability. In semantic category, it is analyzed that research on EPC language with different perspective such as discussion on its syntax, its semantics, identification of OR join problems in EPC and comparison with other BPML is carried out. No study has targeted these four parameters simultaneously but [78], [79] has discussed both the syntax and semantics of EPC. Only one paper [80] has identified the negative impact of OR join on EPC while [79]–[82] has compared EPC with other languages.

F. REQUIREMENT CATEGORY

Requirement gathering is the first step towards business process automation. In few researches, EPC is used in the area of requirement engineering. For example, Gross and Doerr [83] compare the EPC diagram and UML 2 activity diagram in order to detect the effectiveness and efficiency in the domain of requirement engineering. Both languages are used for business modeling. The conclusion states that activity diagrams yields better results for a requirement engineer. The analysis performed in this study is very helpful for both requirement engineers and end users. In another research, Dragicevic *et al.* [84] propose a Method for Elicitation, Documentation and Validation (MeDoV) for requirements. For modeling of requirements, EPC and UML activity diagrams are used. Authors prefer the selection of EPC due to its high acceptance by business users. Later on, the authors demonstrate the applicability of MoDeV framework in requirement engineering [84]. In another study, Groset *et al.* [85] analyze the different languages such as UML activity diagram, use case diagram, EPC, BPMN with Goals, Question and Metrics (GQM) paradigm. Viewpoint of end user and requirement engineer are considered. It is concluded that EPC requires more elements to model the same requirements than BPMN. On the other

TABLE 16. Identified tools for EPC.

Sr. #	Tool Name	Purpose	Reference	Total
Tools Used				
1	ARIS[3]	Modeling	[16][19][44][46][47][49][59][70][71]	9
2	Signavio[4]	Modeling	[76]	1
3	Microsoft Visio	Modeling	[58]	1
4	ADONIS[5]	Modeling, Verification	[19]	1
5	EPCTools[6]	Modeling, Verification	[14] [51] [66]	3
6	Oracle BPA[7]	Transformation,Verification	[41]	1
7	xoEPC[8]	Verification	[46] [47]	2
8	YAWL[9]	Verification	[33][51] [57] [65]	4
9	UPPAAL[10]	Verification	[58]	1
10	TAPAAL[11]	Verification	[58]	1
11	ProM[12]	Verification	[24] [31] [43] [45][46] [47] [50] [51] [52] [53] [54][55] [79]	13
12	Pi-Calculus	Verification	[60]	1
13	CPN Tool	Verification	[25][59]	2
14	WIFA	Verification	[73]	1
Tools Developed				
15	Microsoft Visio (Plug-in)[13]	Modeling	[42]	1
16	CoMoMod	Modeling, Verification	[27]	1
17	EPClets	Verification	[62]	1
18	NetProof	Verification	[48]	1
19	VPD	Integration	[34]	1

hand, BPMN is more errors prone than EPC and use case diagram.

G. EPC TOOLS

We overall identify 19 tools (14 existing and 5 propose tools) from 73 selected papers as presented in **Table 16** with following parameters: 1)**Tool Name** represents the name of the tool 2) **Purpose** represents the functionality of tool in the context of EPC e.g. modeling, transformation, verification etc. 3) **Reference** of the papers is provided for further details. **Table 16** is divided into two sections where the *Tools Used* represents the existing/already developed tools used in the selected studies. On the other hand, *Tools developed* represents the tools that are developed/proposed as a part of research.

From **Table 16**, we analyzed nineteen tools where four tool are used for the **modeling** (i.e. ARIS, Signavio, EPCTools and Visio). The **verification** of EPC is performed by ten tools (i.e. xoEPC, YAWL, UPPAAL, TAPAAL, ProM, Pi-calculus tool, CPN tool, WIFA, EPClets and NetProof). We identify three tools (i.e. ADONIS, EPCTools and CoMoMod) where both modeling and verification of EPC is performed. There is one proposed tool (VPD) to perform EPC integration. Similarly, Oracle BPA is the only tool where both transformation and verification of EPC models is performed.

It can be seen from the **Table 16** that most of the researches used ARIS tool for modeling and ProM tool for verification. EPC was introduced in ARIS for the first time and it is most frequently used tool since then. ProM is an open source tool whose functionality is extended by different researches. New algorithms for analysis/verification of EPC are introduced in ProM with the passage of time. The examples of algorithms include: state space, reduction rules, high level/low level Petri net based verification algorithm and WofAWL. Along with modeling, analysis and verification, Oracle BPA is used for the transformation of EPC to other BPML for analysis/verification.

In tools developed, five tools are customized/developed to achieve different EPC activities which include modeling, integration and verification. In [42], Visio tool is customized through plug-in for error detection. In [62], EPClets tool is developed for verification using eclipse plug-in. In [34], VPD is used for integration of public and private EPC models. In [27], CoMoMod tool is used to perform two activities of EPC; modeling and verification. Also, NetProof in [48] is used for verification of EPC models.

1) ADDITIONAL TOOLS

We identify 19 tools from selected studies as presented in **Table 16**. However, there is a fair chances that we miss few relevant tools not reported in the literature. For example, we selected five scientific repositories to conduct this SLR and it is possible that few tools are reported in the literature which is published by other scientific repositories. Similarly,

TABLE 17. Additional EPC tools.

Sr. #	Name	EPC activity	License type	Support For additional process objects
1	Visual Paradigm[14]	Modeling	Proprietary	Yes
2	Semtalk[15]	Modeling	Microsoft partners	Yes
3	Cubetto[16]	Modeling	Freeware	Yes
4	Edraw Max[17]	Modeling	Freeware	Yes
5	Conceptual Modeling Methods [18]	Modeling, Verification	Freeware	No
6	BflowTooolbox[19]	Modeling, Verification	Eclipse public license	Yes

[3]http://www.ariscommunity.com/aris-express
[4]https://www.signavio.com/
[5]http://en.adonis-community.com/
[6]http://www2.cs.uni-paderborn.de/cs/kindler/research/EPCTools/
[7]http://oracle-business-process-analysis-suite.software.informer.com/11.0/
[8]http://wi.wu-wien.ac.at:8002/epc/
[9]http://yawlfoundation.org/
[10]http://www.uppaal.org/
[11]http://www.tapaal.net/
[12]http://www.promtools.org/doku.php?id=prom651
[13]http://www2.cs.uni-paderborn.de/cs/kindler/research/EPCTools/
[14]https://www.visual-paradigm.com/
[15]http://www.semtation.de/index.php/de/
[16]http://cubetto.semture.de/en/
[17]https://www.edrawsoft.com/download-edrawmax.php
[18]http://www.openmodels.at/web/guest/omilab-in-education/cmmc/download
[19]https://sourceforge.net/projects/bflowtoolbox/files/latest/download?source=files

the industrial tools are usually not reported in literature. Therefore, to perform comprehensive analysis of EPC tools, we search and identify 6 additional tools (**Table 17**) that have been missed during SLR.

We consider four parameters to analyze additional tools as follows: 1) **Name** represents the name of the identified tool with reference 2) **Purpose** represents the functionality of tool in the context of EPC e.g. modeling, transformation, verification etc. 3) **License type** defines the licensing kind of the tool 4) Almost all EPC tools offer the modeling of main elements such as events, functions and connectors. **Additional process objects** represent the support for additional process objects like information object and organizational structure.

It can be seen from the **Table 17** that we identify four modeling tools and two tools to perform both modeling and verification. For example, Visual paradigm provides the modeling of EPC. It is a proprietary tool which also supports the modeling of additional process objects. SemTalk is used for modeling by Microsoft partners and supports additional process objects. Conceptual modeling method is a freeware tool that provides the capabilities of modeling and verification/simulation. In this tool, analysis of EPC models describes the business process, its activities, frequency, execution time, waiting time, resting, transport time, cycle time and cost. BflowToolBox is an eclipse plug-in which provides the functionality of modeling EPC along with additional process objects and it can identify the errors while modeling. The verification of EPC models can be performed by using BflowToolBox at design time. Cubetto offers the modeling of several languages: BPMN, EPC, UML, Flowchart, Mind-map, nodes, edges and business maps. It is the latest tool developed in 2015. The key benefit of this tool is that it provides the layout so modeling is easy and less time consuming as compared to other tools. In EdrawMax, EPC is placed under the business diagrams category where its modeling is available. Both Cubetto and EdrawMax are freeware and supports the modeling of additional process objects.

In addition to identified tools (**Table 17**), we also search and recognize few less significant online tools for EPC. For example, Apromore http://apromore.org/ and BPM academic initiative http://www.signavio.com/bpm-academic-initiative/ are contributing in developing EPC diagrams. Both tools provide the facility of modeling different business process modeling languages such as BPMN, YAWL, EPC and UML. These tools support different formats such as BMPN, YAWL, AML and EPML. In fact BPM academic initiative is globally used for teaching and research purpose.

IV. ANSWERS OF RESEARCH QUESTIONS

RQ1: What are the leading approaches/techniques reported so far to improve the EPC modeling?

Answer: We identified 14 studies (**Table 4**) in modeling category (**Section 3.1**) where significant efforts have been made for the improvement of EPC modeling. In the two studies, various UML diagrams have been exploited to improve EPC modeling (**Table 5**). Moreover, there are 4 studies where EPC notions are extended to improve the modeling capabilities (**Table 6**). Furthermore, we found 3 studies where EPC meta-model is extended to add new modeling constructs (**Table 7**). In addition, we found 5 studies where enhancement of EPC modeling through integration is performed (**Table 8**). Finally, we found 13 tools (**Table 16** and **Table 17**) that support EPC modeling.

It has been analyzed that in UML-based research studies, EPC models are mapped to the different UML structural or behavioral diagrams. The efforts have been made to take advantage of UML in business process modeling. On the other hand, EPC notation based extensions target a particular domain. In meta-model extensions, the EPC meta-model is extended and new elements are introduced and modeled. In addition, we analyzed that there is sufficient tool support available for modeling EPC. Therefore, current EPC approaches certainly improve modeling capacities. It is important to note that it is essential to first develop the models of EPC before verification. Therefore, it can be argued that verification category can be overlapped with modeling category. However, in verification category, only those studies are considered where verification is the primary focus.

RQ2: What are the major transformation strategies to transform EPC models into other target models for further analysis/verification?

Answer: We identified 13 studies (**Table 4**) in transformation category (**Section 3.2**). EPC transformation is further divided into two categories. In first category, 4 studies represent the transformation of EPC to EPC/ EPC extension (**Table 9**). The EPC to EPC transformation enriches the EPC language to efficiently identify the errors. In second category, 9 studies represent the transformation of EPC to other BPML (**Table 10**). The reason to transform EPC language into other BPML is to take advantage of other languages which offers analysis or verification techniques by using their equivalent semantics. We identified one transformation tool in the context of EPC (**Table 16**).

It is analyzed that transformation of EPC to other BPML shows significant results in terms of tool support as compared to transformation of EPC to EPC. However, the transformation of EPC to other types of BPML requires the knowledge of two languages which is an additional effort for the business users.

RQ3: What are the significant methods utilized for the verification of EPC models?

Answer: In verification category (**Section 3.3**), we found total 17 studies (**Table 4**). Verification of EPC involves three types i.e. verification can be done with EPC language (**Table 11**), Petri net language (**Table 12**) and other BPML (**Table 13**). 6 studies are found based on EPC verification, 8 studies represent the Petri net based verification and 3 studies indicate other BPML based verification. We have identified sixteen tools for verification (Table 16 and **Table 17**).

In EPC based verification approaches, the input and output language remains the same i.e. EPC. In this case, ProM tool is frequently utilized for verification. For the early verification at design time, another approach is to convert the EPC language into Petri net. The advantage of doing this conversion is to highlight the importance of Petri net which is a known and formalized language offering the wide range of analysis and verification techniques. The significant tool support and algorithm used for verification of EPC through Petri nets are available. EPC also takes the advantage of other BPML for verification as demonstrated in **Table 13**.

RQ4: What are the leading tools utilized/proposed for the modeling and verification of EPC?

Answer: We overall identified nineteen tools in the literature to perform certain EPC activities (**Table 16**). The identified tools are categorized into exiting tools (14) and proposed/developed tools (5) as presented in (**Table 16**). Furthermore, to perform comprehensive analysis of EPC tools, we search and identify 6 additional tools (**Table 17**) that have been missed during SLR. It has been analyzed that there are adequate tools available especially for modeling and verification of EPC. We found that the ProM is a leading tool for EPC verification. Similarly, ARIS is the foremost tool for the modeling of EPC. Furthermore, some industrial tools are very efficient in modeling such as Cubetto provide modeling layout to simplify and speedup EPC modeling.

RQ5: Is it possible to integrate EPC with other BPML's?

Answer: We have found 5 studies that deals with the integration of EPC language with other languages (**Section 3.1.3**) (**Table 8**). The aim behind this integration is to facilitate the EPC language in a particular domain to handle the complex/ large system requirements. Before integration, two languages need to be used. After integration, only one language is obtained and used. Also, the integration is beneficial for errors checking/verification of EPC models. Therefore, it is concluded that EPC can be integrated with other BPML's to manage the complex business process requirements.

RQ6: Is it possible to model and verify complex/large business requirements through EPC?

Answer: From the SLR, it is analyzed that the primary EPC meta-model is based on atomic events. An atomic event is a single indivisible event which is associated with simple patterns such as parallel synchronization. It is analyzed that atomic events of EPC are unable to model the complex/ large business process requirements. Consequently, there is a strong need to develop an approach to support complex event processing in EPC. We found that 71 selected studies are dealing with the EPC atomic events. However, in only two selected studies [21], [58], the concept of complex events in EPC is proposed which is derived from the novel field of Complex Event Processing (CEP). These complex events provide the basis to include the modeling and verification of complex events in EPC.

Before concluding the answer of this question, it is required to first analyze the atomic and complex events. Therefore, we have performed the comparative analysis of atomic and complex events as given in **Table 18**. Atomic events of EPC have support of workflow patterns such as parallel split, synchronization and exclusive choice while complex event can support the complex event patterns. Marking concept like Petri net is missing in both simple and complex events. Reason for absence of marking concept is that EPC is semi-formal language and marking concepts are present in formal

TABLE 18. Analysis of atomic and complex events for EPC.

Sr. #	Parameters	Atomic Events	Complex Events
1	Support of Complex event patterns	No	Yes
2	Support of workflow patterns	Yes	No
3	Support of Marking concepts	No	No
4	Decision Making	No	Yes
6	Flexibility achieved	No	Yes
5	Modeling available	Full	Partial
7	Verification available	Full	Partial
8	Less concepts	Yes	No

languages like Petri-nets. Some examples of complex event patterns are event location, event time and event cardinality. For atomic events, events cannot make decision thus cannot execute XOR-Split or OR-Split logical operator. While in complex events, events can make decision. An example of complex event is "*using incorrect PIN in Automatic Teller Machine (ATM) more than three times*". As a result of this complex event, decision is made in terms of card capturing of a user. EPC does not execute as it lacks complex events modeling and verification. Complex events in EPC diagrams can be further extended by using meta-model approach and higher flexibility can be achieved. By introducing complex event patterns in EPC, it can be used for real-time modeling and verification. Modeling of complex events is partially available in literature whereas modeling of atomic events is fully present. Also, verification of atomic events is fully supported while verification of two complex event patterns (event time and event trend) is found in only one paper [58].

The modeling of atomic events is fully supported in EPC. Moreover, modeling notations are extended using atomic events of EPC. On the other side, the full modeling support for complex events is not available yet. Furthermore, there are wide number of tool available for atomic events of EPC but there is no tool support available for EPC complex event patterns. Only one study [21] has proposed the modeling of complex events with the help of annotations using EOC notations which can be a good starting point but currently not enough in this area.

Along with modeling of complex events in EPC, its verification should be available to detect errors at design time. There is a significant work available in the literature regarding the verification of atomic events of EPC. However, there is no modeling and verification support in EPC for complex events. Only one study [58] discuss the verification of complex event patterns by mapping event time and event trend complex patterns from EPC to timed automata and time Petri nets. Two verification tools are observed for verification and it is concluded that timed automata is more suitable to verify the atomic and complex events.

On the basis of aforementioned facts, it is concluded that the combination of atomic and complex events are required in EPC to manage large and complex business process requirements. Although the atomic events are well supported in EPC, the inclusion of complex events in EPC is in initial stages and there is no significant approach available yet that support the modeling and verification of complex events in EPC. Consequently, it can be concluded that currently, EPC does not support the modeling and verification of complex and large business process requirements.

V. DISCUSSION

It can be argued that BPMN is more expressive with rich tool support as compared to EPC. Therefore, EPC is now gradually replaced by BPMN or other business modeling languages. However, this is not completely true. The analysis of different business process modeling languages demonstrates that there is no clear winner. Every language has a set of advantages and disadvantages with respect to the domain and business requirements. The key advantages of EPC over BPMN are as follows:

1. BPMN is a new language developed in 2003 while EPC was developed in early nineties. Being old, EPC is used in academia and industry extensively. It is still ongoing subject for discussion whenever business modeling is considered.
2. BPMN supports the execution and verification of the business processes with the help of another language i.e. Business Process Execution Language (BPEL). On the other hand, the primary purpose of EPC is to model the business processes. Execution and verification is achieved by transforming it into other business process modeling languages or formal mechanism. The most commonly used formal mechanism in this context is Petri net.
3. Evaluation of expressive power indicates that EPC is more expressive as compared to BPMN. Here, expressiveness means the less complexity induced while modeling the business processes.
4. The EPC strikes a good balance between severe adherence to rules in modeling and conveying the process in simple terms. EPC has few graphical notations which are simple and easy to understand. This leads to understand the processes to stakeholders which are not even experts in process modeling.
5. According to the effectiveness criteria based on five principles (representational clarity, perceptual discriminability, perceptual immediacy, visual expressiveness, and graphic parsimony), EPC is doing better than BPMN. In fact, EPC proves to be more effective than BPMN because both BPMN and EPC partially support visual expressiveness but EPC completely supports graphic parsimony [87].
6. In any business process, one activity may involve more than one participant. BPMN doesn't support the multiple resources to one activity. On other hand, EPC not only supports this but also offers more than ten different connection types between participant

(organization unit) and activity (function). In terms of resource analysis, EPC is superior to the BPMN.

It can also be argued that EPC diagrammatic representation is not evolving towards a real standardization of modeling capabilities. In this context, two additional references [88] and [89] are provided. According to study [88], there are a number of modeling tools available which implements the modeling languages such as BPMN or EPC. In contrast to BPMN, no efforts to standardize EPC are undertaken yet. Thus, EPC remained as a de facto standard for business process modeling. The major contribution of this paper is the layout to make EPC a standard language. For this purpose, several modeling tools are evaluated. Therefore, in order to standardize EPC, state of art from software point of view is provided and highlighted in this SLR. There is research work [89] where the contribution is made towards standardization of EPC. It is a fact that EPC was developed in 90s and it is widely used in research and practice till now. But due to no accepted standard, different dialects of EPC appeared and disappeared over time. In order to discuss such issues, a review [89] is conducted on exchange formats of EPC. Among seven exchange and storage formats, EPML (EPC Markup Language) is found to have greatest capabilities. Therefore, important step is already taken towards EPC standardization. EPML is part of upcoming standard for the EPC language. In this context, this article also provides a contribution to make EPC a standard language by providing complete EPC developments in a single article.

Although we try to include latest research study in this SLR as much as possible, we are not able to find much studies that are published during 2016-17. The primary reason is that we have selected five renowned databases for this SLR i.e. IEEE, Springer, ACM, Elsevier and Taylor and Francis. Consequently, we perform the search process in only these five databases. There are several other scientific repositories that may contain EPC related studies but credibility of such studies is questionable. Therefore, we do not consider any study which is published in other scientific databases. For example, we found 12 studies [88]–[100] published during 2013-17 in different other scientific repositories. However, we didn't consider such studies in this SLR as we only consider top five renowned databases for search process. We believe that the exclusion of such studies doesn't significantly affect the outcomes of this SLR.

VI. CONCLUSIONS AND FUTURE WORK

This article comprehensively investigates the applications of Event-driven Process Chain (EPC) for the modeling and verification of business requirements. Particularly, a Systematic Literature Review (SLR) is carried out to identify and investigate the 73 research studies, published during 1998-2017, in the domain of EPC. The identified studies are classified into six corresponding categories i.e. Modeling (14), Transformation (13), Verification (17), General (20), Semantic (5), and Requirement (4). Subsequently, each category is thoroughly analyzed to summarize the inclusive EPC developments for the modeling and verification of both simple as well as complex business requirements. Finally, 25 leading EPC tools have been presented i.e. existing tools (14), proposed tools (5) and additional tools (6).

It is concluded from the SLR that the EPC provides adequate approaches and tool support for modeling and verification of simple business requirements through atomic events. However, the complex business requirements cannot be modeled and verified through EPC because it provides very limited modeling and verification support so far for the complex event processing. Consequently, there is a strong need to include the support for modeling and verification of complex events in EPC to manage multifaceted business requirements. Moreover, it is also analyzed that the Petri-nets is the commonly used formalism for the verification of EPC models. In this context, it would be very interesting to investigate other formal methods like timed automata for EPC verification. Furthermore, several languages have been introduced for the modeling and verification of business requirements e.g. BPMN etc. Therefore, it would be required to perform detailed comparative analysis of EPC and BPMN tools. Such comparison will really facilitate practitioners/researchers for selecting the right language and tool as per requirements.

REFERENCES

[1] R. M. Dijkman, M. Dumas, and C. Ouyang, "Semantics and analysis of business process models in BPMN," *Inf. Softw. Technol.*, vol. 50, no. 12, pp. 1281–1294, Nov. 2008.

[2] S. Hinz, K. Schmidt, and C. Stahl, "Transforming BPEL to Petri nets," in *Proc. Int. Conf. Bus. Process Manage.*, 2005, pp. 220–235.

[3] J. G. Molina, M. J. Ortin, B. Moros, J. Nicolas, and A. Toval, "Transforming the OOram three-model architecture into a UML-based process," in *Proc. 34th Annu. Hawaii Int. Conf. Syst. Sci.*, Maui, HI, USA, Jan. 2001, p. 9.

[4] H. Mili, G. Tremblay, G. B. Jaoude, É. Lefebvre, L. Elabed, and G. El Boussaidi, "Business process modeling languages: Sorting through the alphabet soup," *J. Comput. Surv.*, vol. 43, no. 1, pp. 1–56, 2010.

[5] O. Kath, "Towards executable models: Transforming EDOC behavior models to CORBA and BPEL," in *Proc. 8th IEEE Int. Enterprise Distrib. Object Comput. Conf. (EDOC)*, Monterey, CA, USA, Sep. 2004, pp. 267–274.

[6] S. Junginger, "The workflow management coalition standard WPDL: First steps towards formalization," in *Proc. 7th Eur. Concurrent Eng. Conf. (ECEC)*, Leicester, U.K., Apr. 2000, pp. 1–6.

[7] *Business Process and Business Information Analysis Overview. Version 1.0 (ebXML)*, OASIS, Burlington, MA, USA, 2001.

[8] J. Mendling and M. Hafner, "From inter-organizational workflows to process execution: Generating BPEL from WS-CDL," in *Proc. OTM Workshops*, vol. 3762. 2005, pp. 506–515.

[9] *Business Process and Business Information Analysis Overview. Version 1.0 (ebXML)*, OASIS, Burlington, MA, USA, 2001.

[10] R. Mendes and J. P. Vilela, "Privacy-preserving data mining: Methods, metrics, and applications," *IEEE Access*, vol. 5, pp. 10562–10582, 2016.

[11] E. A. Lee, "Constructive models of discrete and continuous physical phenomena," *IEEE Access*, vol. 2, pp. 797–821, Aug. 2014.

[12] E. Söderström, B. Andersson, P. Johannesson, E. Perjons, and B. Wangler, "Towards a framework for comparing process modelling languages," in *Proc. 14th Int. Conf. Adv. Inf. Syst. Eng. (CAiSE)*, Toronto, ON, Canada, 2002, pp. 600–611.

[13] A.-W. Scheer and M. Nüttgens, "ARIS architecture and reference models for business process management," in *Proc. Conf. Bus. Process Manage.*, 2000, pp. 366–379.

[14] N. Cuntz and E. Kindler, "On the semantics of EPCs: Efficient calculation and simulation," in *Proc. 3rd Int. Conf. Bus. Process Manage. (BPM)*, Nancy, France, 2005, pp. 398–403.

[15] B. Korherr and B. List, "A UML 2 profile for event driven process chains," in *Proc. Int. Conf. Res. Practical Issues Enterprise Inf. Syst. (CONFENIS)*, Vienna, Austria, 2006, pp. 161–172.
[16] P. Loos and T. Allweyer, "Object-orientation in business process modeling through applying event driven process chains (EPC) in UML," in *Proc. 2nd Int. Enterprise Distrib. Object Comput. (EDOC)*, 1998, pp. 102–112.
[17] I. Reinhartz-Berger, P. Soffer, and A. Sturm, "Extending the adaptability of reference models," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 40, no. 5, pp. 1045–1056, Sep. 2010.
[18] J. Mendling, G. Neumann, and M. Nüttgens, "Yet another event-driven process chain," in *Proc. 3rd Int. Conf. Bus. Process Manage. (BPM)*, Nancy, France, 2005, pp. 428–433.
[19] M. Rosemann and W. M. P. van der Aalst, "A configurable reference modelling language," *Inf. Syst.*, vol. 32, no. 1, pp. 1–23, 2005.
[20] C. Brandt, F. Hermann, and T. Engel, "Modeling and reconfiguration of critical business processes for the purpose of a business continuity management respecting security, risk and compliance requirements at credit suisse using algebraic graph transformation," in *Proc. 13th Enterprise Distrib. Object Comput. Conf. Workshops*, Sep. 2009, pp. 64–71.
[21] J. Krumeich, N. Mehdiyev, D. Werth, and P. Loos, "Towards an extended metamodel of event-driven process chains to model complex event patterns," in *Proc. Int. Conf. Conceptual Modeling*, Stockholm, Sweden, 2015, pp. 119–130.
[22] B. Korherr and B. List, "Extending the EPC with performance measures," in *Proc. ACM Symp. Appl. Comput. (SAC)*, Seoul, South Korea, 2007, pp. 1265–1266.
[23] V. Stefanov and B. List, "A performance measurement perspective for event-driven process chains," in *Proc. 16th Int. Workshop Database Expert Syst. Appl. (DEXA)*, Aug. 2005, pp. 967–971.
[24] F. Gottschalk, W. M. P. van der Aalst, and M. H. Jansen-Vullers, "Merging event-driven process chains," in *Proc. Confederated Int. Conf.*, Monterrey, Mexico, 2008, pp. 418–426.
[25] Y.-C. Yu, W.-H. Chen, and K.-P. Liu, "Integration of EPC and a modularized colored petri net through events for agile manufacturing cell control," in *Proc. IEEE Int. Conf. Netw., Sens. Control*, Taipei, Taiwan, Mar. 2004, pp. 513–518.
[26] Y.-C. Yu and Y.-T. Hsieh, "A parametric event-driven flowed token petri nets for generic agile systems modeling and control," in *Proc. IEEE Int. Conf. Syst., Man Cybern.*, Taipei, Taiwan, Oct. 2006, pp. 596–601.
[27] T. Dollmann, C. Houy, P. Fettke, and P. Loos, "Collaborative business process modeling with CoMoMod—A toolkit for model integration in distributed cooperation environments," in *Proc. IEEE 20th Int. Workshops Enabling Technol., Infrastruct. Collaborative Enterprises (WETICE)*, Jun. 2011, pp. 217–222.
[28] Y. Liu and D. Wang, "An RFID middleware business process integration framework based on EPC modeling and complex event processing," in *Proc. IEEE 4th Int. Conf. Comput. Sci. Converg. Inf. Technol. (ICCIT)*, Nov. 2009, pp. 64–69.
[29] J. Recker, M. Rosemann, W. van der Aalst, and J. Mendling, "On the syntax of reference model configuration—Transforming the C-EPC into lawful EPC models," in *Proc. Int. Conf. Bus. Process Manage.*, Nancy, France, 2006, pp. 497–511.
[30] J. Mendling, J. Recker, M. Rosemann, and W. van der Aalst, "Generating correct EPCs from configured C-EPCs," in *Proc. ACM Symp. Appl. Comput. (SAC)*, Dijon, France, 2006, pp. 1505–1510.
[31] M. Murzek, G. Kramler, and E. Michlmayr, "Structural patterns for the transformation of business process models," in *Proc. 10th IEEE Int. Enterprise Distrib. Object Comput. Conf. Workshops (EDOCW)*, Oct. 2006, p. 18.
[32] W. Ling, J. Wang, and J. Yan, "Analysis of mapping form EPC model to workflow model," in *Proc. Int. Technol. Innov. Conf. (ITIC)*, Nov. 2007, pp. 1854–1857.
[33] J. Mendling, M. Moser, and G. Neumann, "Transformation of yEPC business process models to YAWL," in *Proc. ACM Symp. Appl. Comput. (SAC)*, Dijon, France, 2006, pp. 1262–1266.
[34] J. Ziemann, D. Werth, T. Matheis, and T. Kahl, "Towards tool support for integrated modeling of private and public business processes," in *Proc. 16th IEEE Int. Workshops Enabling Technol., Infrastruct. Collaborative Enterprises (WETICE)*, Jun. 2007, pp. 383–386.
[35] V. Hoyer, E. Bucherer, and F. Schnabel, "Collaborative e-business process modelling: Transforming private EPC to public BPMN business process models," in *Proc. Int. Conf. Bus. Process Manage. (BPM)*, Brisbane, QLD, Australia, 2008, pp. 185–196.
[36] M. Strommer, M. Murzek, and M. Wimmer, "Applying model transformation by-example on business process modeling languages," in *Proc. Int. Conf. Conceptual Modeling-Found. Appl.*, Auckland, New Zealand, 2007, pp. 116–125.
[37] O. Levina, "Assessing information loss in EPC to BPMN business process model transformation," in *Proc. IEEE 16th Int. Enterprise Distrib. Object Comput. Conf. Workshops (EDOC)*, Sep. 2012, pp. 51–55.
[38] A. Winter and C. Simon, "Using GXL for exchanging business process models," *Inf. Syst. e-Bus. Manage.*, vol. 4, no. 3, pp. 285–307, 2005.
[39] O. Thomas, K. Leyking, and F. Dreifus, "Using process models for the design of service-oriented architectures: Methodology and E-commerce case study," in *Proc. 41st Annu. Hawaii Int. Conf. Syst. Sci.*, Waikoloa, HI, USA, Jan. 2008, p. 109.
[40] D. Vanderhaeghen and P. Loos, "Process modelling approach for collaboration networks," in *Proc. 15th IEEE Int. Workshops Enabling Technol., Infrastruct. Collaborative Enterprises (WETICE)*, Jun. 2006, pp. 263–264.
[41] L. O. Meertens, M.-E. Iacob, and S. M. Eckartz, "Feasibility of EPC to BPEL model transformations based on ontology and patterns," in *Proc. Int. Conf. Bus. Process Manage. (BPM)*, Berlin, Germany, 2010, pp. 347–358.
[42] M. Smuts, C. Burger, and B. Scholtz, "Composite, real-time validation for business process modelling," in *Proc. Southern African Inst. Comput. Sci. Inf. Technol. (SAICSIT)*, 2014, p. 93.
[43] B. F. van Dongen, A. K. A. de Medeiros, H. M. W. Verbeek, A. J. M. M. Weijters, and W. M. P. van der Aalst, "The ProM framework: A new era in process mining tool support," in *Proc. Int. Conf. Appl. Theory Petri Nets (ICATPN)*, Miami, FL, USA, 2005, pp. 444–454.
[44] J. Mendling and W. van der Aalst, "Formalization and verification of EPCs with OR-joins based on state and context," in *Proc. Int. Conf. Adv. Inf. Syst. Eng. (CAiSE)*, Trondheim, Norway, 2007, pp. 439–453.
[45] J. Mendling, "Verification of EPC Soundness," in *Metrics for Process Models* (Lecture Notes in Business Information Processing), vol. 6. Berlin, Germany: Springer, 2008, pp. 59–102.
[46] J. Mendling, G. Neumann, and W. van der Aalst, "Understanding the occurrence of errors in process models based on metrics," in *On the Move to Meaningful Internet Systems: CoopIS, DOA, ODBASE, GADA, and IS* (Lecture Notes in Computer Science), vol. 4803. Berlin, Germany: Springer, 2007,
[47] J. Mendling, G. Neumann, and W. van der Aalst, "On the correlation between process model metrics and errors," in *Proc. 26th Int. Conf. Conceptual Modeling*, Auckland, New Zealand, 2007, pp. 173–178.
[48] P. Langner, C. Schneider, and J. Wehler, "Petri net based certification of event-driven process chains," in *Proc. 19th Int. Conf. Appl. Theory Petri Nets (ICATPN)*, Lisbon, Portugal, 1998, pp. 286–305.
[49] W. M. P. van der Aalst, "Formalization and verification of event-driven process chains," *Inf. Softw. Technol.*, vol. 41, no. 10, pp. 639–650, 1999.
[50] W. M. P. van der Aalst*et al.*, "ProM 4.0: Comprehensive support for real process analysis," in *Proc. 28th Int. Conf. Appl. Theory Petri Nets Other Models Concurrency (ICATPN)*, Siedlce, Poland, 2007, pp. 484–494.
[51] V. Gruhn and R. Laue, "A comparison of soundness results obtained by different approaches," in *Proc. Int. Conf. Bus. Process Manage. (BPM)*, Berlin, Germany, 2010, pp. 501–512.
[52] J. Mendling, B. F. van Dongen, W. M. P. van der Aalst, "Getting rid of the OR-join in business process models," in *Proc. 11th IEEE Int. Enterprise Distrib. Object Comput. Conf. (EDOC)*, Oct. 2007, p. 3.
[53] N. Lohmann, E. Verbeek, and R. Dijkman, "Petri net transformations for business processes—A survey," in *Transactions on Petri Nets and Other Models of Concurrency II* (Lecture Notes in Computer Science), vol. 5460. Berlin, Germany: Springer, 2009, pp. 46–63.
[54] J. Mendling, B. F. van Dongen, and W. M. P. van der Aalst, "Getting rid of OR-joins and multiple start events in business process models," *Enterprise Inf. Syst.*, vol. 2, no. 4, pp. 403–419, 2008.
[55] B. F. van Dongen, W. M. P. van der Aalst, and H. M. W. Verbeek, "Verification of EPCs: Using reduction rules and Petri nets," in *Proc. 17th Int. Conf. Adv. Inf. Syst. Eng. (CAiSE)*, Porto, Portugal, 2005, pp. 327–386.
[56] B. F. Van Dongen, J. Mendling, W. M. P. Van Der Aalst, "Structural patterns for soundness of business process models," in *Proc. 10th IEEE Int. Enterprise Distrib. Object Comput. Conf. (EDOC)*, Oct. 2006, pp. 116–128.

[57] J. Mendling, H. M. W. Verbeek, B. F. van Dongen, W. M. P. van der Aalst, and G. Neumann, "Detection and prediction of errors in EPCs of the SAP reference model," *Data Knowl. Eng.*, vol. 64, no. 1, pp. 312–329, 2008.
[58] A. Amjad, F. Azam, M. W. Anwar, and W. H. Buttm, "Verification of event-driven process chain with timed automata and time Petri nets," in *Proc. 9th IEEE GCC Conf. Exhib.*, Bahrain, May 2017.
[59] K. van Hee, O. Oanea, and N. Sidorova, "Colored Petri nets to verify extended event-driven process chains," in *Proc. OTM Confederated Int. Conf. 'Move Meaningful Internet Syst.'*, 2005, pp. 183–201.
[60] W. Xia, L. Lan, and M. Junpeng, "Research on flexible business process of bank modeling based on EPC," in *Proc. IEEE 24th Int. Conf. Manage. e-Commerce e-Government (ICMeCG)*, Oct./Nov. 2014, pp. 54–60.
[61] T. Dong, H. Cai, and B. Xu, "A business process modeling approach based on semantic event-driven process chains," in *Proc. 14th Int. Conf. Comput. Supported Cooperative Work Design (CSCWD)*, Apr. 2010, pp. 201–206.
[62] M. Kapuruge, J. Han, A. Colman, and U. Rüegg, "EPClets—A lightweight and flexible textual language to augment EPC process modelling," in *Proc. IEEE Int. Conf. Services Comput. (SCC)*, Jun./Jul. 2014, pp. 693–700.
[63] J. Mendling, "Event-driven process chains (EPC)," in *Metrics for Process Models* (Lecture Notes in Business Information Processing), vol. 6. Berlin, Germany: Springer, 2008, pp. 17–57.
[64] J.-H. Pfeiffer, W. R. Rossak, and A. Speck, "Applying model checking to workflow verification," in *Proc. 11th IEEE Int. Conf. Workshop Eng. Comput.-Based Syst. (ECBS)*, May 2004, pp. 144–151.
[65] W. M. P. van der Aalst, M. Dumas, F. Gottschalk, A. H. M. ter Hofstede, M. La Rosa, and J. Mendling, "Preserving correctness during business process model configuration," *Formal Aspects Comput.*, vol. 22, no. 3, pp. 459–482, 2009.
[66] E. Kindler, "On the semantics of EPCs: Resolving the vicious circle," *J. Data Knowl. Eng.*, vol. 56, no. 1, pp. 23–40, 2005.
[67] G. Decker and J. Mendling, "Instantiation semantics for process models," in *Proc. 6th Int. Conf. Bus. Process Manage. (BPM)*, Milan, Italy, 2008, pp. 164–179.
[68] S. Xue, B. Wu, and J. Chen, "lightEPC: A formal approach for modeling personalized lightweight event-driven business process," in *Proc. IEEE Int. Conf. Services Comput. (SCC)*, Jun./Jul. 2013, pp. 1–8.
[69] S. Mei, H. Cai, and F. Bu, "Multi-view service-oriented rule merged business process modeling framework," in *Proc. IEEE 6th Int. Symp. Service Oriented Syst. (SOSE)*, Dec. 2011, pp. 175–180.
[70] A. Bögl, M. Schrefl, G. Pomberger, and N. Weber, "Automated construction of process goal trees from EPC-models to facilitate extraction of process patterns," in *Proc. 11th Int. Conf. Enterprise Inf. Syst. (ICEIS)*, Milan, Italy, 2009, pp. 427–442.
[71] S. Feja and D. Fötsch, "Model checking with graphical validation rules," in *Proc. 15th Annu. IEEE Int. Conf. Workshop Eng. Comput. Based Syst. (ECBS)*, Mar./Apr. 2008, pp. 117–125.
[72] D. Irfan, X. Xiaofei, and D. Shengchun, "Business process modeling for SCM systems," in *Proc. 4th Int. Conf. Emerg. Technol. (ICET)*, Rawalpindi, Pakistan, Oct. 2008, pp. 253–257.
[73] A. Tsai, J. Wang, W. Tepfenhart, and D. Rosca, "EPC workflow model to WIFA model conversion," in *Proc. IEEE Int. Conf. Syst., Man Cybern.*, Taipei, Taiwan, Oct. 2006, pp. 2758–2763.
[74] J. Mendling and M. Nüttgens, "EPC markup language (EPML): An XML-based interchange format for event-driven process chains (EPC)," *Inf. Syst. e-Bus. Manage.*, vol. 4, no. 3, pp. 245–263, 2005.
[75] S. D. Sivri and H. Krallmann, "Process-oriented knowledge management within the product change systems of the automotive industry," *J. Procedia Eng.*, vol. 100, pp. 1032–1039, Feb. 2015.
[76] K. Yongsiriwit, N. Assy, and W. Gaaloul, "A semantic framework for configurable business process as a service in the cloud," *J. Netw. Comput. Appl.*, vol. 59, pp. 168–184, Jan. 2016.
[77] D. M. M. Schunselaar, J. Gulden, H. van der Schuur, and H. A. Reijers, "A systematic evaluation of enterprise modelling approaches on their applicability to automatically generate ERP software," in *Proc. IEEE 18th Conf. Bus. Inform. (CBI)*, Paris, France, Aug./Sep. 2016, pp. 290–299.
[78] P. S. Santos, Jr., J. P. A. Almeida, and G. Guizzardi, "An ontology-based semantic foundation for ARIS EPCs," in *Proc. ACM Symp. Applied Comput.*, Cham, Switzerland, 2010, pp. 124–130.
[79] B. van Dongen, R. Dijkman, and J. Mendling, "Measuring similarity between business process models," in *Proc. 20th Int. Conf. Adv. Inf. Syst. Eng. (CAiSE)*, Montpellier, France, 2008, pp. 450–464.
[80] K. Sarshar and P. Loos, "Comparing the control-flow of EPC and Petri net from the end-user perspective," in *Proc. 3rd Int. Conf. Bus. Process Manage. (BPM)*, 2005, pp. 434–439.
[81] G. Grossmann, M. Schrefl, and M. Stumptner, "Modelling inter-process dependencies with high-level business process modelling languages," in *Proc. 5th Asia–Pacific Conf. Conceptual Modelling (APCCM)*, vol. 79. 2008, pp. 89–102.
[82] G. Jošt, J. Huber, M. Heričko, and G. Polančić, "An empirical investigation of intuitive understandability of process diagrams," *Comput. Standards Interfaces*, vol. 48, pp. 90–111, Nov. 2016.
[83] A. Gross and J. Doerr, "EPC vs. UML activity diagram—Two experiments examining their usefulness for requirements engineering," in *Proc. IEEE 17th Int. Conf. Requirement Eng.*, Aug./Sep. 2009, pp. 47–56.
[84] S. Dragicevic, S. Celar, and L. Novak, "Use of method for elicitation, documentation, and validation of software user requirements (MEDoV) in agile software development projects," in *Proc. IEEE 6th Int. Conf. Comput. Intell., Commun. Syst. Netw. (CICSyN)*, May 2014, pp. 65–70.
[85] A. Gross, J. Jurkiewicz, J. Doerr, and J. Nawrocki, "Investigating the usefulness of notations in the context of requirements engineering," in *Proc. IEEE 2nd Int. Workshop (EmpiRE)*, Chicago, IL, USA, Sep. 2012, pp. 9–16.
[86] S. Dragicevic and S. Celar, "Method for elicitation, documentation and validation of software user requirements (MEDoV)," in *Proc. IEEE 6th CICSyN*, 2013, pp. 65–67.
[87] K. Figl, J. Mendling, M. Strembeck, and J. Recker, "On the cognitive effectiveness of routing symbols in process modeling languages," in *Proc. 13th Int. Conf. Bus. Inf. Syst.*, May 2010, pp. 230–241.
[88] D. M. Riehle, S. Jannaber, A. Karhof, P. Delfmann, O. Thomas, and J. Becker, "Towards an EPC standardization—A literature review on exchange formats for EPC models," in *Proc. Conf. Multikonferenz Wirtschaftsinformatik (MKWI)*, 2016, pp. 1168–1178.
[89] D. M. Riehle, S. Jannaber, A. Karhof, O. Thomas, P. Delfmann, and J. Becker, "On the de-facto standard of event-driven process chains: How EPC is defined in literature," in *Modellierung* (Lecture Notes in Informatics). Bonn, Germany: Köllen Druck+Verlag GmbH, 2016.
[90] K. Grzybowska and G. Kovács, "The modelling and design process of coordination mechanisms in the supply chain," *J. Appl. Logic*, vol. 24, pp. 25–38, Nov. 2017.
[91] D. Jugel, S. Kehrer, C. M. Schweda, and A. Zimmermann, "Providing EA decision support for stakeholders by automated analyses," *Digital Enterprise Computing* (Lecture Notes in Informatics). Bonn, Germany: Gesellschaft für Informatik e.V, 2015.
[92] J. O. Sá, C. Kaldeich, and J. Á. Carvalho, "A multi-driven approach to requirements analysis of data warehouse model: A case study," *IADIS Int. J. Comput. Sci. Inf. Syst.*, vol. 8 no. 1, pp. 14–30, 2013.
[93] P. Ardalani, C. Houy, P. Fettke, and P. Loos, "Towards a minimal cost of change approach for inductive reference model development," in *Proc. ECIS*, 2013, pp. 1–12.
[94] M. Radloff, M. Schultz, and M. Nüttgens, "Extending different business process modeling languages with domain specific concepts: The case of internal controls in EPC and BPMN," in *Enterprise Modelling and Information Systems Architectures* (Lecture Notes in Informatics). Bonn, Germany: Gesellschaft für Informatik, 2015.
[95] V. Repa and T. Bruckner, "Methodology for modeling and analysis of business processes (MMABP)," *J. Syst. Integr.*, vol. 6, no. 4, pp. 17–28, 2015.
[96] I. Chishti, J. Ma, and B. Knight, "Ontology mapping of business process modeling based on formal temporal logic," *Int. J. Adv. Comput. Sci. Appl.*, vol. 5, no. 7, pp. 15–95, 2014.
[97] C. Müller and C. Krüger, "A tool for a flexible posterior analysis of simulation experiments," in *Proc. 29th Eur. Conf. Modelling Simulation*, 2015, pp. 1–6.
[98] H. Kern, "Model interoperability between meta-modeling environments by using M3-level-based bridges," Ph.D. dissertation, Faculty Math. Comput. Sci., Univ. Leipzig, Leipzig, Germany, 2016.

[99] C. Müller, "Approaches to run simulations of business processes in a grid computing network," in *Proc. 28th Eur. Conf. Modelling Simulation*, 2014, pp. 1–4.
[100] K. Grigorova and K. Mironov, "Bridging the gap between different interfaces for business process modeling," *Int. J. Comput. Inf. Eng.*, vol. 9, no. 12, pp. 2479–2482, 2015.

ANAM AMJAD received the B.S. degree in computer sciences from Fatima Jinnah Women University, Rawalpindi, Pakistan. She is currently pursuing the M.S. degree in software engineering with the Computer Engineering Department, College of Electrical and Mechanical Engineering, National University of Sciences and Technology, Pakistan. Her area of research is business process automation through model driven software engineering.

FAROOQUE AZAM is currently an adjunct faculty member with the Department of Computer Engineering, College of Electrical and Mechanical Engineering, National University of Sciences and Technology, Pakistan. He has been teaching various software engineering courses since 2007. His areas of interests are model driven software engineering, business modeling for Web applications, and business process reengineering.

MUHAMMAD WASEEM ANWAR is a Senior Researcher and Industry Practitioner in the field of embedded/control systems. He is currently pursuing the Ph.D. degree with the Computer and Software Engineering Department, College of Electrical and Mechanical Engineering, National University of Sciences and Technology, Pakistan. His major research areas are model based system engineering for embedded systems, supervisory control and data acquisition systems, and natural language processing for design automation.

WASI HAIDER BUTT is currently an Assistant Professor with the Department of Computer Engineering, College of Electrical and Mechanical Engineering, National University of Sciences and Technology, Pakistan. His areas of interests are model driven software engineering, and Web development and requirement engineering.

MUHAMMAD RASHID received the bachelor's degree in electrical engineering from the University of Engineering and Technology at Peshawar, Peshawar, Pakistan, in 2000, the master's degree in embedded systems design from the University of Nice Sophia Antipolis, France, in 2006, and the Ph.D. degree in embedded systems design from the University of Western Brittany, Brest, France, in 2009. He is currently an Assistant Professor with the Computer Engineering Department, Umm Al-Qura University, Makkah, Saudi Arabia.