Enterprise Architecture Modeling Support based on Data Extraction from Business Process Models

Dmytro Orlovskyi [0000-0002-8261-2988] and Andrii Kopp [0000-0002-3189-5623]

National Technical University "KhPI", Kyrpychova str. 2, 61002 Kharkiv, Ukraine
orlovskyi.dm@gmail.com, kopp93@gmail.com

Abstract. This paper presents a problem of enterprise architecture artifacts extraction from business process model collections, which organizations of higher maturity levels tend to manage, in order to build the architecture landscape and apply enterprise architecture management practices. Existing enterprise architecture frameworks, languages, and methodologies are discussed. Proposed approach is based on mapping between business process and enterprise architecture elements in order to present all business architecture artifacts in a single coarse-grained model. The software implementation allows generating business architecture landscapes that could be used for architecture evolution purposes, such as transformation planning or maintenance efforts evaluation.

Keywords: Enterprise Architecture, Business Process Model, Business Architecture, Architecture Landscape, ArchiMate.

1 Introduction

These days Enterprise Architecture (EA) has two definitions upon the context. It can be considered as a formal description of a system or a detailed plan of the system at component level to guide its implementation. On the other hand, EA may be defined as the structure of components, their interrelations, principles and guidelines governing their design and evolution over time [1]. The term "enterprise" denotes any organization linked by a common set of goals. Despite existing misconceptions, EA is covering not only operational software systems, but strategic, business, and organizational aspects as well. Therefore, EA deals with requirements and strategies, as well as business processes, technical applications, and infrastructures. EA strives for optimal articulation between these different facets [2].

The EA is expressed using models, which are based on a metamodel that defines model types and their relationships. Various EA frameworks have their own particular metamodel and taxonomy, but in general, all the EA frameworks cover the following domains [1, 3, 4]:

- Business Architecture: business goals, business functions or capabilities, and business processes. It defines the business strategy, governance, organization, and key business processes.

- Data Architecture: data stores and data objects mapped to the business functions and business processes that need specific data. It describes the structure of an organization's logical and physical data assets and data management resources.
- Application Architecture: the structure and behavior of software applications that produce and consume data used by business functions and business processes. It provides a blueprint for the individual application systems to be deployed, their interactions, and their relationships to the core business processes of the organization.
- Technology Architecture: the structure and behavior of the IT (Information Technology) infrastructure (client/server nodes, system software, protocols and networks). It describes the logical software and hardware capabilities that are required to support the deployment of business, data, and application services.

As it was already mentioned, EA is often positioned only within the context of IT governance. However, EA is actually related with a number of well-known best practices and standards in IT and general management [4]. The set of the best practices, frameworks, guidelines, and standards outlined below serves to support each of the considered EA viewpoints (business, data, applications, and technical architectures):

- Strategic Management: Balanced Scorecard (BSC). The scorecard measures organizational performance across four perspectives: financial, customers, business processes, learning and growth. The BSC enables companies to track financial results while monitoring progress in building the capabilities and acquiring intangible assets they need for future growth [5].
- Business model development: Business Model Canvas. A Business Model Canvas (BMC) helps users visually represent the elements of a business model and the potential interconnections and impacts on value creation. As a visual tool, the BMC can facilitate discussion, debate, and exploration of potential innovations to the underlying business model itself; with users developing a more systemic perspective of an organization and highlighting its value creating impacts [6].
- Business architecture: BIZBOK (Business Architecture Body of Knowledge) and O-BA (Open Business Architecture). BIZBOK defines business architecture as a blueprint of the enterprise that provides a common understanding of the organization and is used to align strategic objectives and tactical demands [7].
- Quality Management: EFQM (European Foundation for Quality Management) and ISO (International Organization for Standardization) 9001. The EFQM model has a much broader scope than ISO 9001, since it not only focuses on quality management, but provides an overall management framework for performance excellence of the entire organization [4].
- IT governance: COBIT (Control Objectives for Information and Related Technologies) framework that assists enterprises in achieving their objectives for the governance and management of enterprise IT. It helps enterprises create optimal value from IT by managing a balance between realization benefits and optimization risk levels and resource use [8].
- IT delivery and support: ITIL (IT Infrastructure Library). The ITIL is a technique to manage the technology and communications in an optimal way. The primary ob-

jective of the ITIL is to establish the best practices and improving the standard of IT service quality that customers should demand and providers should supply [9].

- IT implementation: CMM (Capability Maturity Model) and CMMI (Capability Maturity Model Integration). CMMI contains practices that cover project management, process management, systems engineering, hardware engineering, software engineering, and other supporting processes used in development and maintenance of both products and services [10].

In order to organize these diverse EA perspectives into a holistic and unified view, it is required to use an Enterprise Architecture Framework (EAF) [11]. According to reference [1], EAF should describe a method for designing an information system in terms of a set of building blocks, and for showing how the building blocks fit together. However, EA is not only about information systems. Thus, the goal of EAF is to provide a language, an approach, and a set of recommendations covering all facets of the EA, from organization and strategy, to business and technology, to planning and change management [2].

Modern EA frameworks define the Architecture Landscape (AL) as the representation of EA assets that are planned or already in use by the enterprise. The AL contains models of the existing architecture across the entire enterprise. Such models deal with business processes, applications, and data. Naturally, AL content is constantly evolving as architectural transformation take place. All the EA content, such as models and other architectural descriptions, is contained in the EA repository that serves as the source of EA models for the AL [2]. It is necessary to understand, that the EA repository is not concrete software or data storage but just an abstract concept of a single place where the enterprise descriptions should be stored for their further reuse for EA evolution purposes. Also, gathering the information about all or the most valuable EA assets and their preparing according to the practiced EAF might be a long-term and quite expensive project. However, many organizations maintain repositories of business process models that serve as a knowledge base for their ongoing business process management efforts [12].

Hence, in this paper we present the idea on how the EA artifacts might be extracted from the collection of business process models in order to build the AL and apply EA management practices. Processing of a large business process model collection might help to retrieve information about business processes and business functions, corresponding inputs and outputs, participants of business processes, and triggering events. This approach is supposed to shorten time, save costs and efforts for gathering information in order to design the business architecture view of the whole AL, which then might be complemented with required data, applications, and technical architectures artifacts that support business process execution.

2 Literature Review

Many EA methodologies have come and gone in the last decades. According to [13], most enterprises use one of these four EAFs: The Zachman Framework for Enterprise Architecture, The Open Group Architecture Framework (TOGAF), Federal Enterprise

Architecture Framework (FEAF), and the Gartner Enterprise Architecture Framework (GEAF). Additionally to these enterprise frameworks, the latest research [11] considers the United States Department of Defense Architecture Framework (DoDAF) and derived from it NATO (North Atlantic Treaty Organization) Architecture Framework (NAF), British Ministry of Defense Architecture Framework (MODAF), and Unified Architecture Framework (UAF) responding to the needs of military communities to create a standardized and consistent EA based on DoDAF and MODAF frameworks.

Originally, the first EAF was designed by John Zachman. He described the EAF as a logical structure for classifying and organizing the descriptive representations of an enterprise that are significant to the management of the enterprise, as well as to the development of the enterprise's systems [14]. The Zachman Framework is actually the taxonomy for organizing architectural artifacts (design documents, specifications, EA models etc.), which takes into account both who the artifact targets (e.g., business owner or another stakeholder) and what particular issue (e.g., data and functionality) is being addressed [13]. Thus, authors of [15] have formalized the Zachman Framework using the ontology model. Detailed overview of the Zachman Framework for Enterprise Architecture (ZFEA) is described in [16]. One of the authors of this paper is John Zachman, the originator of the framework. They have described the ZEFA as a bounded matrix with six rows and six columns. The columns are the interrogatives, answers to which allow describing all aspects of any enterprise, while the rows represent different perspectives on the enterprise from the viewpoint of different stakeholders [16]. Besides the Zachman Framework, military frameworks (DoDAF, NAF, MODAF, and UAF), and the United Stated federal methodology FEAF, TOGAF was identified in [11] as the best EAF, mostly according to the criteria: information availability, tool support, and prevalence by researchers. In the overview of architectural frameworks authors of [17], together with DoDAF and FEAF, mentioned another U.S. federal EA methodology, such as Treasury Enterprise Architecture Framework (TEAF). However, they declared only TOGAF and the Zachman Framework as the two popular EAFs that are used nowadays [17].

TOGAF is the one of the leading EA frameworks worldwide. It is developed and is currently maintained as a standard by The Open Group (TOG). The TOGAF documents focus on EA key concepts and Architecture Development Method (ADM), an iterative approach to developing the EA [18]. The ArchiMate EA modeling language is a TOG standard as well; it provides the architect with instruments that support and improve the architecture process [18]. The generic ArchiMate metamodel consists of two main types of elements: structure and behavior elements. Structure elements can be subdivided into active structure elements and passive structure elements. Active structure elements can be further subdivided into external active structure elements (also called interfaces) and internal active structure elements. Behavioral elements can be subdivided into internal behavior elements, external behavior elements (also called services), and events [19].

The ArchiMate specification also declares its relationship to other standards, specifications, and guidance documents (see Fig. 1). Among these standards are TOGAF, BIZBOK, BPMN, and UML (Unified Modeling Language) [19]. BPMN stands for Business Process Model and Notation. It is the leading standard for modeling business processes introduced by the Object Management Group (OMG). Due to its popularity, BPMN is considered to be the true lingua franca of Business Process Management (BPM). By allowing BPM practitioners to create business process models using a common graphical notation, BPMN makes it easier to communicate processes in a compact way across companies and continents [20]. Latest BPTrends survey [21] declares that BPMN process models are used by 64% of questioned organizations.

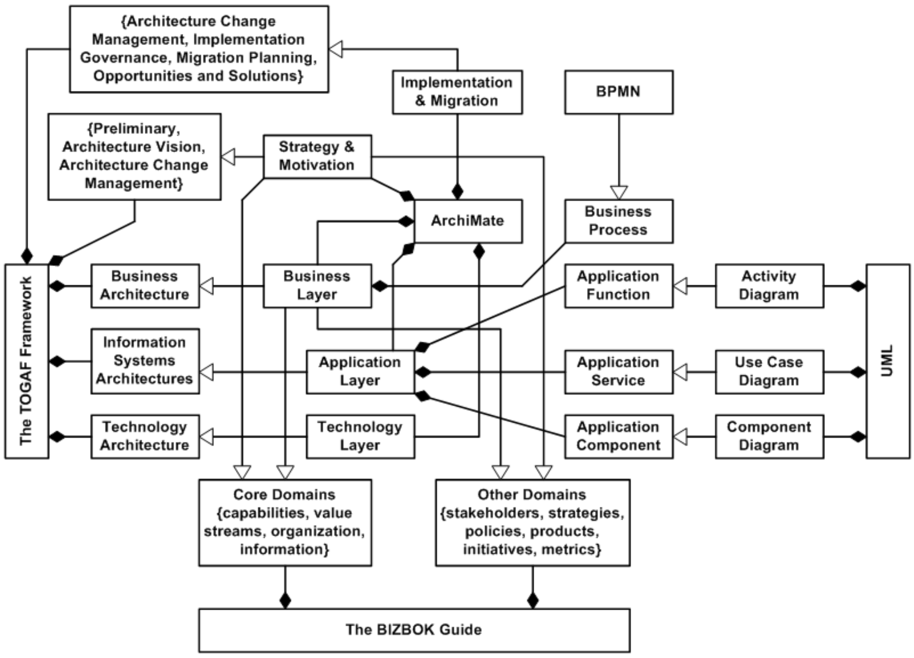

Fig. 1. ArchiMate relationship to other standards

Both ArchiMate and BPMN can be used for modeling business processes, but their aims are different. ArchiMate is used for high-level processes and their relations to the enterprise context, whereas BPMN offers detailed workflow modeling, but lacks the application services that support a process or goals it has to fulfill [19]. Therefore, BPMN business process models might be used as the source of artifacts only for the BA perspective of the EA landscape according to a bottom-up approach.

Authors of the paper [22] have proposed the way, in which the ArchiMate Business Layer and BPMN meta-models can be linked. However, this paper does not provide any direct mapping between corresponding ArchiMate and BPMN elements. Author of [23] proposes mapping of ArchiMate elements to corresponding BPMN elements. However, the linkage between BPMN and ArchiMate defined at the metamodel level is not outlined in details and, therefore, cannot be checked or proven. More detail and cogent mapping is proposed in [24]. Author of this paper have performed the syntactic, semantic, and structural analysis of ArchiMate and BPMN meta-models. However, mapping results presented in paper [24] contain many-to-many relations between ArchiMate and BPMN elements, which cause uncertainty. Thus, it is

ArchiMate and BPMN elements, which cause uncertainty. Thus, it is required to define one-to-one linkage between ArchiMate and BPMN elements by matching metamodels of the considered standards.

3 Formal Problem Statement

ArchiMate was chosen for description of the EA, since it is a contemporary, open and independent language. It comprises three main modeling layers: business, application, and technology. ArchiMate allows presenting a whole EA in the form of views which, depending on the needs, can include only items in one layer or can show vertical relations between layers. The internal structure of an ArchiMate model constructs a graph of nodes linked by directed edges. Both nodes and edges are attributed with information indicating a type of element or relation [25].

Formally ArchiMate EA model can be represented using the tuple (1) [25]:

$$AM = \langle V, E, C, R, vt, et \rangle. \quad (1)$$

Here V – the set of vertices; $E \subset V \times V$ – the set of edges; C – the set of element types; R – the set of relations; $vt : V \rightarrow C$ – the function that assigns element types to graph vertices; $et : E \rightarrow R$ – the function that assigns relation types to edges.

In order to provide the AL template in ArchiMate language, it is required to define a set of vertices, where each vertex $v \in V$ describes the EA asset of a certain type. To do this, it is necessary to define the mapping between BPMN and ArchiMate artifacts. The software implementation of the proposed approach should take the collection of BPMN models as input and produce the ArchiMate model with pre-defined EA elements of the business layer ready to be used for AL design purposes as output. The conceptual scheme of a considered problem is demonstrated in Fig. 2.

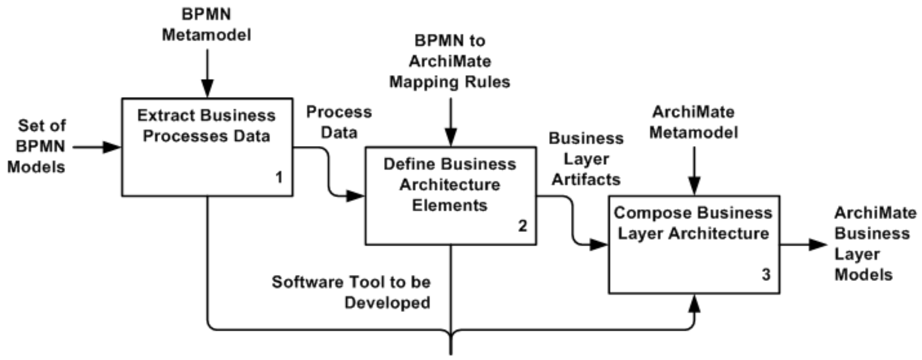

Fig. 2. Conceptual scheme of the considered problem

Such bottom-up approach might save time and resources required to design the EA by providing a ready-to-use ArchiMate document with pre-defined business architecture elements (Fig. 2).

4 Proposed Approach

In order to define mapping of BPMN elements to ArchiMate elements, we have came up with the idea to use the corresponding meta-models of these standards. Such meta-models are provided in paper [22] and their fragments are depicted in Fig. 3.

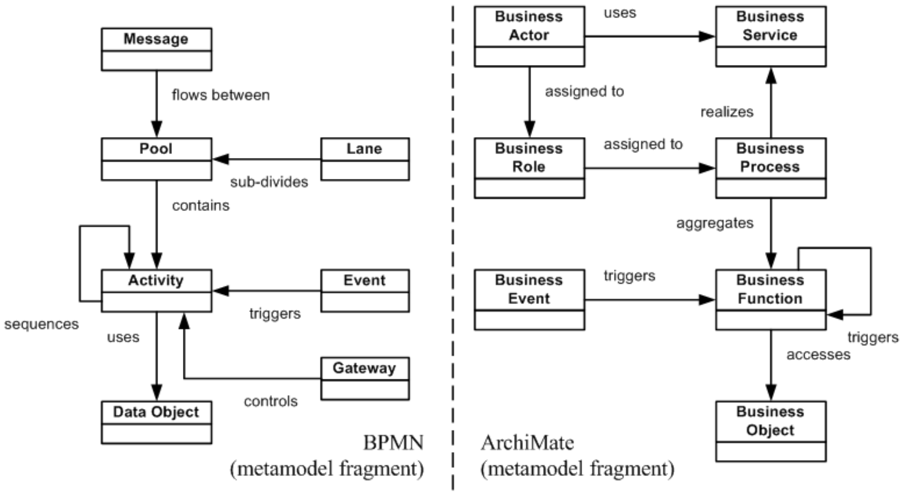

Fig. 3. Fragments of BPMN and ArchiMate Business Layer meta-models

We have translated such meta-models into a collection of RDF (Resource Description Framework) statements (provided in a form of "subject-predicate-object" triples), originally designed to represent a metadata better than other relational or ontological models [26]. As the result, we obtained RDF-graphs that have been queried in order to define similarities and then to conclude mapping rules between their nodes. For this purpose we have used Apache Jena, the open-source Java-based framework used in Linked Data and Semantic Web applications [27]. It provides API (Application Programming Interface) that allows querying RDF models directly or with the help of the query language SPARQL.

The starting point is our awareness that a business event in ArchiMate is similar to BPMN event elements, which is outlined in the ArchiMate specification [19]. Since in BPMN an event triggers an activity object, we can conclude that in ArchiMate a business event also triggers some sort of a behavior element. It is shown that SPARQL is an SQL-like language, so queries used to define relations between the event and other elements are the following:

```
SELECT * WHERE { <Event> ?a ?x . }
SELECT * WHERE { <Business Event> ?b ?y . }
```

Results of these SPARQL queries are shown in Fig. 4. Therefore, it is seen, that the BPMN activity might be represented as the business function in ArchiMate.

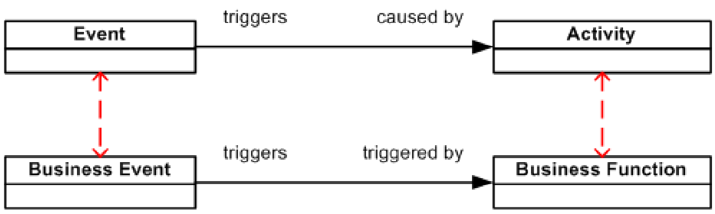

Fig. 4. Matching BPMN Activity to ArchiMate Business Function

Then, we can check the "Activity" and "Business Function" nodes of BPMN and ArchiMate meta-models respectively:

```
SELECT * WHERE { <Activity> ?a ?x . }
SELECT * WHERE { <Business Function> ?b ?y . }
```

Query results of outgoing arcs in RDF-graph have shown that both activity and business function have cycles ("sequences" and "triggers" relations) used to model a flow of activities. Also both activity and business function are related to data objects in BPMN and business objects in ArchiMate. These relations have labels with similar terms "uses" and "accesses", so we can assume that BPMN data objects could be mapped to ArchiMate business objects (Fig. 5).

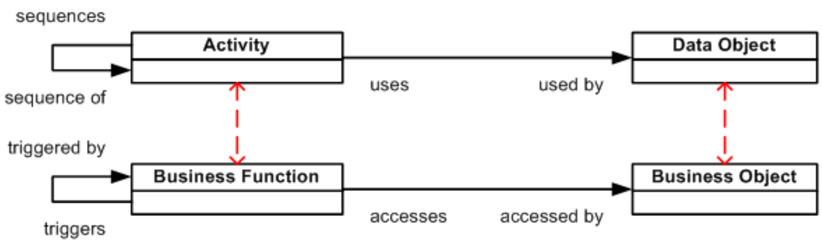

Fig. 5. Matching BPMN Data Object to ArchiMate Business Object

In order to analyze incoming arcs we have used the following SPARQL queries:

```
SELECT * WHERE { ?x ?a <Activity> . }
SELECT * WHERE { ?y ?b <Business Function> . }
```

Query results of incoming arcs in RDF-graph have shown that both activity and function are parts of (by analyzing relationships "contains" and "aggregates", which meaning is very close) other elements "Pool" and "Business Process" of BPMN and ArchiMate meta-models respectively. Also it is known that a pool in BPMN sets the boundaries of a business process. Hence, we can assume that BPMN pools could be mapped to ArchiMate business processes (Fig. 6).

Incoming arcs of "Pool" in the BPMN metamodel and "Business Process" in the ArchiMate metamodel were defined using the following queries:

```
SELECT * WHERE { ?x ?a <Pool> . }
SELECT * WHERE { ?y ?b <Business Process> . }
```

Fig. 6. Matching BPMN Flow Objects to ArchiMate Internal Behavior Elements

Lanes in BPMN are used to organize tasks of a business process according to roles responsible for performing these tasks, so that is why we can assume that "Lane" of BPMN could be mapped to "Business Role" of ArchiMate (Fig. 7).

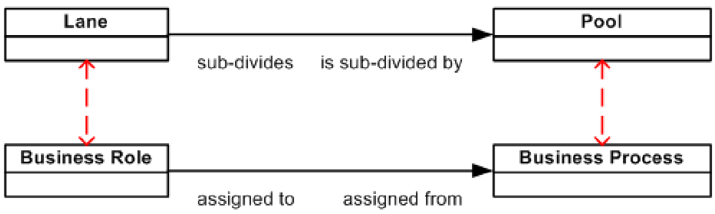

Fig. 7. Matching BPMN Lane to ArchiMate Business Role

Performed analysis is based on meanings of objects of both BPMN and ArchiMate meta-models and relations between such objects. All of this manual work might be formalized on the basis of a breadth-first search (BFS) algorithm [28]. Therefore, we have modified the original BFS algorithm so it could be applied for traversing RDF-graphs that represent considered BPMN and ArchiMate meta-models:

```
RDF-BFS(start_node):
  visited = [start_node]
  queue = [start_node]
  queried = []                  # already queried triples
  while !queue.empty():
    node = queue.poll()
    objects = execute(SELECT * WHERE { node ?x ?y . })
    while statement = objects.next():
      if !queried.contains(statement):
        queried.add(statement)  # mark queried outgoing
                                # relation as traversed
```

```
      if !visited.contains(?y):
        queue.add(?y)
        visited.add(?y)
  subjects = execute(SELECT * WHERE { ?x ?y node . })
  while statement = subjects.next():
    if !queried.contains(statement):
      queried.add(statement)   # mark queried incoming
                               # relation as traversed
      if !visited.contains(?x):
        queue.add(?x)
        visited.add(?x)
```

Unlike the classic BFS algorithm, the proposed modification uses the additional list "queried" in order to memorize already queried RDF-triples. It is also harder than in ordinary graphs to get the adjacent edges of a certain vertex. For this purpose we need to execute two SPARQL queries in order to retrieve outgoing and incoming relations separately (see the pseudo code above). Using the RDF-BFS algorithm it is possible to traverse more complicated BPMN and ArchiMate ontologies in order to support the BPM-EA interoperability. Results of BPMN and ArchiMate meta-models traversing are shown in Table 1.

Table 1. Results of BPMN and ArchiMate meta-models traversing

Node	Statements (BPMN)	Node	Statements (ArchiMate)
Event	Event triggers Activity	Event	Event triggers Function
Activity	Activity uses Data Object, Activity sequences Activity, Gateway controls Activity, Pool contains Activity	Function	Function accesses Object, Function triggers Function, Process aggregates Function
Pool	Lane sub-divides Pool, Message flows between Pool	Process	Process realizes Service, Role assigned to Process

Obtained results (Table 1) correspond to the results of manual RDF-graphs querying, which proves concluded mapping rules (Fig. 4-7). By analyzing results of BPMN and ArchiMate meta-models traversing (i.e. correspondence between nodes and relations) we have defined one-to-one linkage between ArchiMate and BPMN elements.

5 Experiments and Results

Developed software solution is based on the provided mapping rules between BPMN and ArchiMate elements. Its workflow is described using the UML activity diagram that is shown in Fig. 8. Business process models described using the BPMN notation are usually stored in a form of the XML-based (eXtensible Markup Language) language called BPMN 2.0. Its structure like of any other XML document consists of nodes and attributes, which represent elements and sequence flows within a business process. With the help of a BPMN model API [29] we can easily extract information from an existing business process definition, edit an existing business process definition or create a complete new one without manual XML parsing.

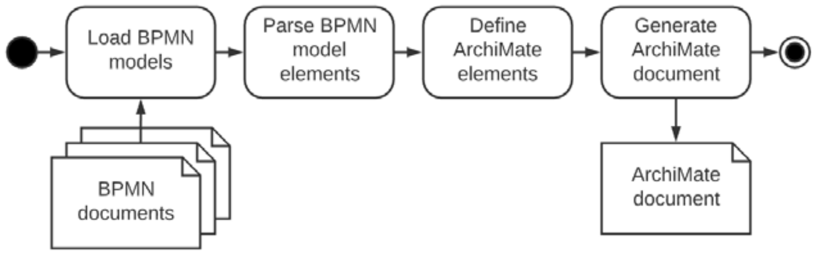

Fig. 8. Activity diagram that represents a workflow of the software solution

As for ArchiMate, its exchange file format is defined by The Open Group Standard and can be used to exchange data between tools and/or systems that wish to import, and export ArchiMate models [30]. The ArchiMate exchange file format is also based on XML language and supported by multiple EA modeling tools.

The UML deployment diagram in Fig. 9 demonstrates the architecture of the software solution. The Java-based web application, created using the Spring Boot framework, loads BPMN 2.0 files from a catalog on the FTP (File Transfer Protocol) server and parses business process definitions using the BPMN model API [29]. Then, for each element of a business process extracted from the BPMN 2.0 file, a corresponding BA artifact (business process, function, event, etc.) is defined according to the mapping rules provided in previous section (Fig. 4-7).

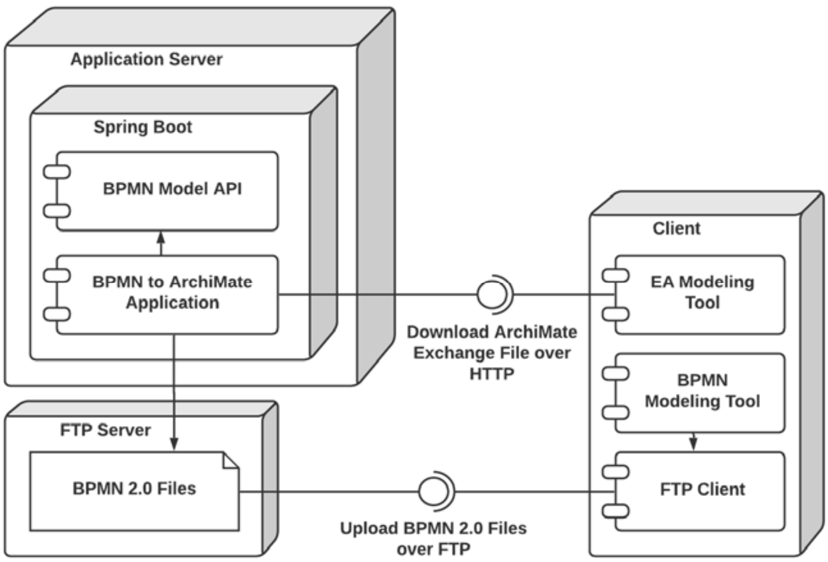

Fig. 9. Architecture of the software solution

On output the application generates an XML document formed according to the ArchiMate exchange file format. It could be downloaded over the HTTP (HyperText Transfer Protocol) manually or directly accessed by any EA modeling tool that supports the ArchiMate exchange file format.

Sample set of BPMN models that describe business processes in the field of products supply was used to validate proposed solution (Fig. 10).

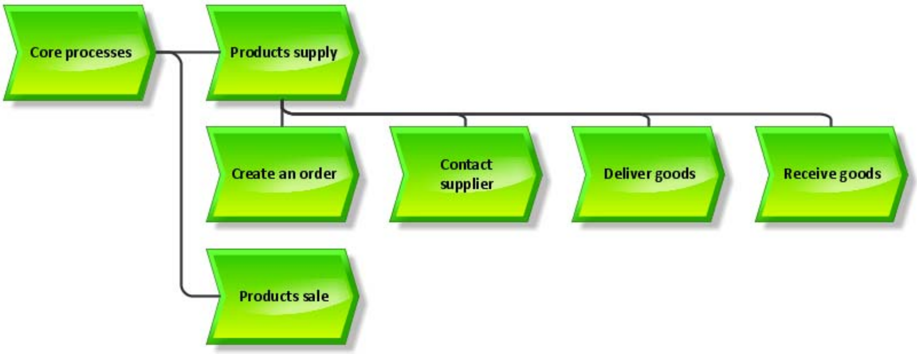

Fig. 10. Generic business process landscape

Since we are limited in space, let us demonstrate a single BPMN model of a whole set, which describes "Create an order" business process (Fig. 11). However, the set of considered business process models in BPMN 2.0 format could be found at [31].

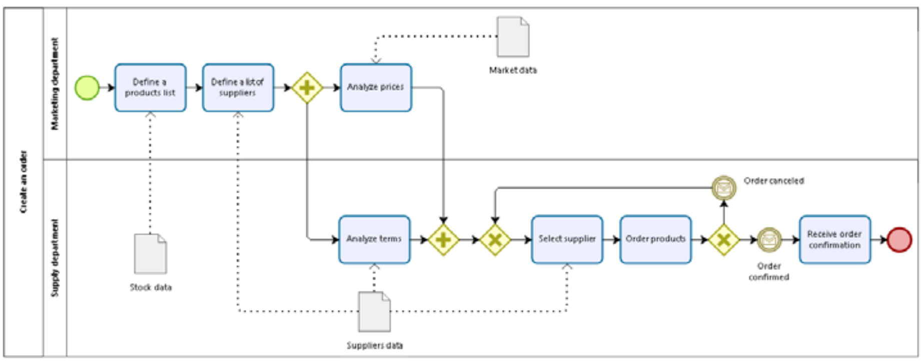

Fig. 11. "Create an order" business process model

Complete BA landscape for the given business processes (Fig. 10) is demonstrated in Fig. 12. As the EA modeling tool we have used Archi, which is open source, cross-platform solution that also supports the ArchiMate exchange file format. As the result, we have obtained business architecture elements that form the set of vertices V that could be augmented by relations (composition, assignment, triggering etc.) that form the set R according to the formal definition AM of the EA model. Besides relations, the set of BA elements could be extended by adding data, applications, and technical architecture elements, and relations.

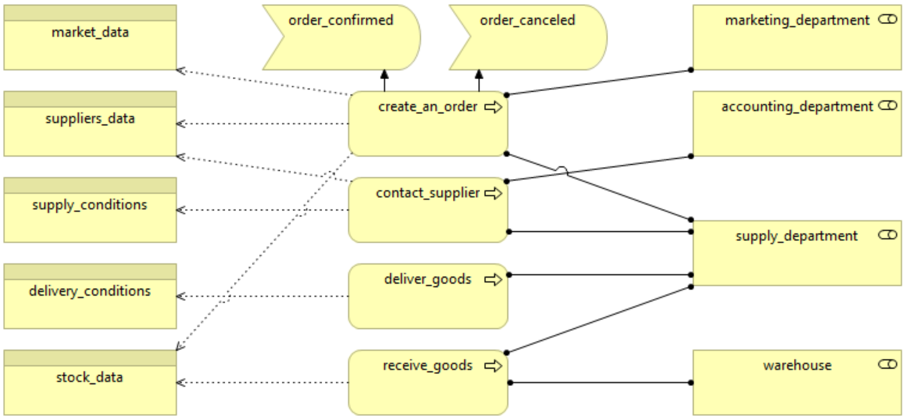

Fig. 12. Business architecture landscape generated from the given process models

Business functions are avoided on the presented EA model (Fig. 12) in order to achieve clearer and less congested diagram. It is shown that all labels are presented in lower case with space characters replaced with underscores in order to avoid duplicated strings (e.g., the "Create an order" label of the pool in the BPMN diagram is transformed into the "create_an_order" label of the business process in the EA model). In order to evaluate the obtained business architecture landscape, we used the Propagation Cost (2), which equals to the fraction of the architecture affected when a change is made to a randomly selected element [32]:

$$\text{PropagationCost} = \frac{1}{|V|^2} \cdot \sum_{v \in V} \deg^+(v) = \frac{1}{|V|^2} \cdot \sum_{v \in V} \deg^-(v). \qquad (2)$$

It is computed from $\deg^+(v)$ – the fan-in (the number of elements that depend on a certain element) or $\deg^-(v)$ – the fan-out (the number of elements that a certain element depends on) [32]. Obtained propagation cost (0.10) means that only 10% of the business architecture may be affected in case of required transformation.

6 Conclusion and Future Work

In this study we have discussed the problem of EA artifacts extraction from the collection of business process models, which may exists in BPM-mature organizations, in order to build the architecture landscape and apply EA management practices. We have proposed the approach based on ArchiMate and BPMN metamodels comparison used to define direct mapping between business process and EA modeling elements of business architecture layer. In contrast to existing studies in this field, such as [23, 24], we have provided one-to-one mapping between BPMN and ArchiMate modeling elements (Fig. 4-7), which is based on the meaning of nodes and relations of the corresponding meta-models.

Proposed approach was formalized using the BFS algorithm extended by the RDF-querying features. Developed software that implements proposed approach is web-based and interoperable, since it supports BPMN 2.0 and ArchiMate exchange formats. Provided example demonstrates the business architecture landscape extracted from the set of BPMN models. In order to demonstrate utility of the EA landscape, we evaluated EA maintenance efforts and evolvability with the help of the propagation cost measure. However, instead of or together with the propagation cost there might be used another link analysis metrics in order to evaluate obtained business architecture landscape.

Future work includes elaboration in the field of automatic EA modeling, which can be referred as "EA-mining" similarly to the process mining in BPM field.

References

1. Pieterse, R.: Enterprise Architecture Frameworks, Methods and Tools. Lulu.com (2015)
2. Desfray, P., Raymond, G.: Modeling Enterprise Architecture with TOGAF: A Practical Guide Using UML and BPMN. Morgan Kaufmann (2014)
3. Jensen, C. T., Cline O., Owen M.: Combining Business Process Management and Enterprise Architecture for Better Business Outcomes. IBM Redbooks (2011)
4. Lankhorst, M.: Enterprise Architecture at Work: Modelling, Communication and Analysis. Springer (2017)
5. Pramudita, C. D.: The Balanced Scorecard as Strategic Controlling Instrument. Introducing the Indicators-based BSC for Implementation of a Corporate Strategy from Four Different Perspectives. Anchor Academic Publishing (2016)
6. Joyce, A., Paquin, R. L.: The triple layered business model canvas: A tool to design more sustainable business models. Journal of cleaner production, vol. 135, p. 1474-1486 (2016)
7. Swenson, K. et. al.: Business and Dynamic Change: The Arrival of Business Architecture. Future Strategies Inc. (2019)
8. COBIT 5: A Business Framework for the Governance and Management of Enterprise IT. ISACA (2012)
9. Mora, M., Gómez, J. M., Garrido, L., Pérez, F. C.: Engineering and Management of IT-based Service Systems: An Intelligent Decision-Making Support Systems Approach. Springer Science & Business Media (2013)
10. Chrissis, M. B., Konrad, M., Shrum, S.: CMMI for Development: Guidelines for Process Integration and Product Improvement. Pearson Education (2011)
11. Bankauskaite, J.: Comparative analysis of enterprise architecture frameworks. CEUR Workshop Proceedings, vol. 2470, p. 61-64 (2019)
12. Dijkman, R., Dumas, M., Van Dongen, B., Käärik, R., Mendling, J.: Similarity of business process models: Metrics and evaluation. Information Systems, vol. 36, no. 2, p. 498-516 (2011)
13. Sessions, R.: A comparison of the top four enterprise-architecture methodologies. Microsoft Developer Network Architecture Center, p. 1-31 (2007)

14. Zachman, J. A.: The framework for enterprise architecture: background, description and utility. Zachman International, p. 1-5 (1996)
15. Lee, J. Y., Ahn, K., Lee, S.: Applying OWL Ontology to Zachman Framework for Requirement Analysis. Advanced Science and Technology Letters, vol. 46, p. 34-37 (2014)
16. Lapalme, J., Gerber, A., Van der Merwe, A., Zachman, J., De Vries, M., Hinkelmann, K. Exploring the future of enterprise architecture: A Zachman perspective. Computers in Industry, vol. 79, p. 103-113 (2016)
17. Raj, P., Raman, A., Subramanian, H.: Architectural Patterns: Uncover essential patterns in the most indispensable realm of enterprise architecture. Packt Publishing (2017)
18. Vicente, M., Gama, N., da Silva, M. M.: Using ArchiMate and TOGAF to understand the enterprise architecture and ITIL relationship. International Conference on Advanced Information Systems Engineering. Springer, Berlin, Heidelberg, p. 134-145 (2013)
19. ArchiMate 3.1 Specification. Generic Metamodel. https://pubs.opengroup.org/architecture/archimate3-doc/chap04.html
20. Bonnet, F., Decker, G., Dugan, L., Kurz, M., Misiak, Z., Ringuette, S.: Making BPMN a true lingua franca. BPM Trends (2014)
21. Harmon, P.: The State of Business Process Management 2016. BPTrends (2016)
22. van den Berg, M., Solutions, R. I.: ArchiMate, BPMN and UML: An approach to harmonizing the notations. Orbus, software, white paper (2012)
23. Penicina, L.: Linking BPMN, ArchiMate, and BWW: Perfect match for complete and lawful business process models?. CEUR Workshop Proceedings, vol. 1023, p. 156-165 (2013)
24. Gill, A. Q., Qureshi, M. A.: Adaptive Enterprise Architecture Modelling. JSW, vol. 10, no. 5, p. 628-638 (2015)
25. Klimek, R., Szwed, P.: Verification of ArchiMate process specifications based on deductive temporal reasoning. IEEE 2013 Federated Conference on Computer Science and Information Systems, pp. 1109-1116 (2013)
26. Resource Description Framework (RDF). https://www.w3.org/RDF/
27. Apache Jena. https://jena.apache.org/getting_started/
28. Even, S.: Graph Algorithms. Cambridge University Press (2011)
29. BPMN Model API. https://docs.camunda.org/manual/7.7/user-guide/model-api/bpmn-model-api/
30. ArchiMate Model Exchange File Format. http://www.opengroup.org/xsd/archimate/
31. BPMN models. https://github.com/andriikopp/research-calculations/tree/master/research-calculations-java/processModelsStorage/Supply_process_BPMN
32. Lagerström, R., Baldwin, C., MacCormack, A., Dreyfus, D.: Visualizing and Measuring Enterprise Architecture: An Exploratory BioPharma Case. IFIP Working Conference on The Practice of Enterprise Modeling. Springer, Berlin, Heidelberg, p. 9-23 (2013)