Towards the Enterprise Architecture Web Mining Approach and Software Tool

Andrii Kopp and Dmytro Orlovskyi

National Technical University "Kharkiv Polytechnic Institute", Kyrpychova str. 2, Kharkiv, 61002, Ukraine

Abstract

This paper considers the enterprise architecture model extraction from organizational websites in an automatic way to simplify the blueprinting of enterprise architecture landscapes at the conceptual level. Thus, such a technique is proposed to be called "enterprise architecture web mining". Nowadays almost all organizations offer their products and services through their websites, therefore, representing their value-creating business processes on the Internet. Thus, enterprise homepages can be considered as sources of business information sufficient to understand the company's business processes landscape and make further decisions depending on the party that uses such information. The proposed approach includes two major stages of business activity detection using hyperlinks of the company's webpage that could represent triggers of certain e-commerce business processes, and enterprise architecture model creation based on the obtained data. The software implementation of the proposed approach uses natural language processing to detect business activities on the corporate web pages and produces human-readable enterprise architecture models that describe business processes offered by examined organizations and supportive application and technology environment. Obtained models represent knowledge about primary business activities conducted by organizations and could be used for decision-making. As the result, the enterprise architecture landscapes were built for several organizations using only their publicly available websites. The limitations are discussed, the conclusion is made, and future work in this field is formulated.

Keywords

Enterprise Architecture, Business Process, Web Content Mining, Natural Language Processing

1. Introduction: Related Work and Problem Statement

Among various definitions of Enterprise Architecture (EA), the main narrative is that EA describes the structure and organization of a company [1]. The two general definitions of EA say that it is can be considered as follows [1]:

- the organizing logic for business processes and information technology (IT) infrastructure that reflects integration and standardization requirements of the operating model used by an enterprise;
- the conceptual blueprint that defines the structure and operations of a company, and determines how it can achieve its ongoing and future goals in the most efficient way.

Unlike the information system architecture, EA describes four architecture domains that describe a whole organization: business architecture, data architecture, applications architecture, and technology architecture (or sometimes referred as the IT infrastructure) [2]. Inside each of the mentioned domains, the detailed descriptions and interconnections between EA artifacts are given [3]:

- business architecture includes business processes, events, services, roles, actors, etc.;
- data architecture includes data objects, entities, attributes, etc.;

Information Technology and Implementation (IT&I-2022), November 30 - December 02, 2022, Kyiv, Ukraine
EMAIL: kopp93@gmail.com (A. Kopp); orlovskyi.dm@gmail.com (D. Orlovskyi)
ORCID: 0000-0002-3189-5623 (A. Kopp); 0000-0002-8261-2988 (D. Orlovskyi)

- applications architecture includes application components, services, interfaces, etc.;
- technology architecture includes system software, nodes, devices, artifacts, etc.

Using EA blueprints, organizations can understand the efficiency of movement toward current and future objectives and make decisions on necessary changes to improve efficiency. Moreover, EA gives a general overview of a whole system, even the large and complex ones. Using the EA approach an organization can define gaps between the ongoing and desired states using various viewpoints, define initiatives that should be implemented to achieve the future state, and continuously track the EA changes over time toward the planned state. The evolution of EA is always defined by its business domain – business processes and services they realize to offer the organization's external or internal consumers dictate the necessary landscape of software systems and IT infrastructure. In their turn, business processes, services, and created products depend on the organization's goals and capabilities. Therefore, EA could be considered as a structured high-level description of an organization from different viewpoints (i.e. business, data, applications, and technology [3]) that serve each other in a layered bottom-up manner. This paper proposes an approach and a software tool for the automatic extraction of EA landscapes from websites that nowadays virtually represent organizations on the Internet. This approach aims at simplifying the procedure of building high-level models in the preliminary stages of EA development. It is well known that today most enterprises offer their products and services on their homepages top-ranked by multiple search engines.

Usually, organizational websites contain information not only about offered products or services but also about related activities that allow customers to receive respective products or services (e.g. order, buy, learn, etc.). The study object is the procedure of EA structure extraction from organizational websites that serve as virtual enterprise representations on the Internet. The study subject is the approach and software tool to extract EA landscapes from organizational websites. The study goal is to simplify the process of EA description in the early stages of EA development. This paper is organized in the following way. In the next subsections, EA frameworks and modeling approaches are discussed, virtual enterprise representation on the Internet is considered, and a formal problem statement is given. In Section 2 the proposed approach to the automatic EA construction based on organizational websites is outlined. Section 3 includes the description of a developed software tool, analysis, and discussion of obtained results. Section 4 contains a conclusion and formulates future work in this field.

1.1. Related Work

1.1.1. Enterprise Architecture Frameworks

The origins of EA refer back to the late 80s when J. Zachman introduced the paper "A Framework for Information Systems Architecture". When the so-called Zachman Framework (ZF) was proposed, organizations had much simpler information systems landscapes than they have today. Thus, with time the ZF was updated and used not only as of the information systems framework but as the Enterprise Architecture Framework (EAF) across various organizations [4].

A "framework" is the term usually met in the software development field. It is considered as the set of building blocks that help developers to provide generic capabilities of a software solution. The software development frameworks tend to provide ready source code that only should be customized or extended to satisfy the particular software requirements. Such source code could be given in the form of libraries, toolkits, application programming interfaces (API), etc. [5]. The EAF concept also uses the framework principles mentioned above, but to set an organization, not only the software system. Existing EA frameworks tend to provide general recommendations and reference solutions that may help in creating and managing EA. EA frameworks also suggest the form of EA description (i.e. models, documents, blueprints, matrices, etc.) [5]. Except the ZF, which has lost its relevance to the modern business processes and IT infrastructures, the most popular EA frameworks are:

- The Open Group Architecture Framework (TOGAF) – an EA framework created and supported by The Open Group that provides a detailed methodology and tools for EA development; its core Architecture Development Method (ADM) provides enterprises with a detailed approach to step-by-step EA development [6];

- Federal Enterprise Architecture Framework (FEAF) – a complex framework by the Federal Government of the United States that is focused on developing and maintaining the EA capabilities; it provides a standardized method and principles for creating and exchanging EA information among Federal agencies [6];
- Department of Defense Architecture Framework (DoDAF) – an architecture framework that is intended to help systems engineers to describe complex systems; it is emerged in the United States Department of Defense as the structure for EA development for engineering and acquisition staff to describe the whole system [7];
- Ministry of Defense Architecture Framework (MoDAF) – an EAF adapted and extended by the United Kingdom Ministry of Defense from the DoDAF; the unique MoDAF viewpoints added to the original DoDAF include strategic and acquisition views to describe high-level requirements for enterprise change and programmatic details respectively [7].

However, the TOGAF is still the most popular EA framework because of its constant development over the last two decades to become an EA development standard [8].

1.1.2. Enterprise Architecture Modeling

The ArchiMate EA modeling language is also authored by The Open Group, authors of TOGAF. This language provides a visual notation to illustrate enterprise architecture elements and relationships between EA elements in a standardized way. Besides the EA domains, this powerful language allows modeling stakeholders, requirements, goals, etc. [9].

ArchiMate describes business processes, including their structure and flows, organizational structure elements, application systems, information flows, and technology infrastructure (Table 1). The goal of ArchiMate modeling is to provide a tool to depict changes in EA elements and relationships, evaluate the decision consequences, and communicate EA solutions [10].

1.1.3. Enterprise Architecture Web Mining: State-of-the-Art

As was given in the introduction section, the suggested “EA web mining” technique is focused on the automatic construction of EA models using corporate websites as sources of data about EA elements and the relationships between them. Hence, the main problem is finding mentions of business processes and other EA elements in HyperText Markup Language (HTML) pages of corporate websites. Whereas the direct search in Google Scholar using the “enterprise architecture web mining” key phrase did not give any results, the “enterprise architecture mining” allowed us to discover several studies in this direction:

- in [11] the author states that manual maintenance of EA models is costly and time-consuming, so they propose EA mining algorithms and tools based on process mining;
- the study [12] also considers automatic EA modeling methods that are supposed to reduce the drawbacks of manual EA modeling (error-proneness, time and cost consumption, accuracy, etc.);
- the systematic review [13] also states that automatic EA modeling could respond challenges of manual EA modeling but this field is still immature and requires further research.

1.2. Problem Statement

The formal representation of an ArchiMate EA model is the following [14]:

$$AM = \langle V, E, C, R, vt, et \rangle, \tag{1}$$

where:

- V is the set of vertices that represent EA model elements;
- $E \subset V \times V$ is the set of edges that represent relationships between EA model elements;
- C is the set of ArchiMate element types;
- R is the set of ArchiMate relationship types;
- $vt: V \rightarrow C$ is the mapping between ArchiMate element types and graph vertices;
- $et: E \rightarrow R$ is the mapping between ArchiMate relationship types and graph edges.

Table 1
ArchiMate elements [10]

Domains	Passive structure elements	Behavior elements	Active structure elements
Business	Business objects	Business services, functions, processes	Business actors and roles
Application	Data objects	Application services, functions, processes	Application components and interfaces
Technology	Artifacts	Technology services, functions, processes	Devices, System software, Communication networks

Hence, the AM tuple (1) should be automatically constructed using the HTML web page tags, their attributes, and inner text fragments. First of all, the web page should be parsed to work with its tags, their attributes, and text content. Formally it can be given using the following equation:

$$PT = Parse(URL), \tag{2}$$

where:

- URL is the Uniform Resource Locator (URL) of a web page that should be parsed;
- PT is the set of tags obtained after the web page parsing;
- $Parse: URL \rightarrow PT$ is the function that defines a mapping between URL addresses of web pages URL and parsed tags PT that belong to these web pages.

Then web page tags obtained using (2) should be used to extract the data about the organization's activity described on its web page on the Internet. The following formalism describes this step:

$$TD = Extract(PT), \tag{3}$$

where:

- $TD = \{tag = \langle tagName, attr: attrName \rightarrow attrValue, text\rangle\}$ is the bag of web page tags tag, each of which has a name $tagName$, attributes $attr$ (whose values $attrValue$ are accessible through their names $attrName$), and a text content $text$;
- $Extract: PT \rightarrow TD$ is the function that defines a mapping between web page parsed tags PT and structured tag data TD elements.

Using the structured tag data obtained using (3), business activities that help an organization virtually promote its products or services on the Internet should be detected. Formally this operation could be described using the following equation:

$$BA = Detect(TD), \tag{4}$$

where:

- BA is the set of business activities detected after the processing of the set of structured tag data elements TD;
- $Detect: TD \rightarrow BA$ is the function that defines a mapping between structured tag data TD elements and business activities BA.

Finally, using the set of business activities obtained using (4) and the previous outcomes, the EA model should be built using the following formalism:

$$AM = Build(URL, Title, Desc, BA), \tag{5}$$

where $Build: \langle URL, Title, Desc, BA\rangle \rightarrow AM$ is the function that defines a mapping between URL addresses of web pages URL, web page title $Title$ and description $Desc$ meta tags content, and business activities BA on the one side and the ArchiMate EA model AM on the other side.

The conceptual model of automatic EA model construction using the company's homepage on the Internet, based on introduced transformations (2) – (5), is demonstrated in Fig. 1.

The proposed workflow (Fig. 1) should help automatically build high-level architectural models using only the websites of organizations using the suggested technology we can name "enterprise architecture web mining". Obtained models may describe landscapes of top-level business processes based on products or services offered to customers on the company's homepage. Moreover, obtained EA models should include application layers to demonstrate website maps, and technology layers to complete the ArchiMate cross-layer architecture. However, the most valuable outcome is still a business architecture layer that includes core value-added business processes and the business service offered to the organization's clients. ArchiMate EA models automatically produced using the company's website can help to understand the current state of the enterprise, including its customer relationship strategy, offered products, and services. Then, shortcomings could be detected in such an EA model, and the decisions to improve the enterprise's virtual representation on the Internet could be made.

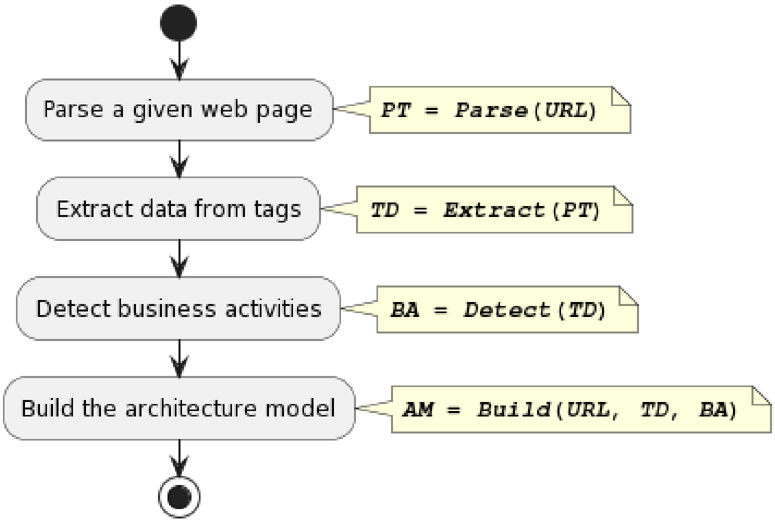


Figure 1: The conceptual model of automatic web-based EA model construction or "EA web mining"

2. Proposed Approach

2.1. Business Activities Detection in Organizational Web Pages

The first HTML tags that should be processed using the proposed approach are "title" and "meta". These tags contain descriptive information about a web page and, therefore, about the organization and products or services it virtually offers on the Internet.

The text content of the "title" tag can be obtained by processing the structured tag data TD elements in the following way based on tuple calculus formalisms [15]:

$$Title \in tag, tag \in \{tag: \{text\} | tag \in TD \land tag.tagName = "title"\}, \tag{6}$$

where $Title$ is the web page title data.

Then it is proposed to process the "meta" tag, which "name" attribute has the value "description" to get the value of its "content" attribute. This could be formally described using the following equation based on tuple calculus formalisms [15]:

$$\begin{gathered} Desc \in tag, tag \in \{tag: \{tag.attr("content")\} | tag \in TD \land tag.tagName = \text{meta} \land \\ \land\, tag.attr("name") = "description"\}, \end{gathered} \tag{7}$$

where $Desc$ is the web page description data.

We propose to use the web page description as the "Business service" ArchiMate element to reflect the product(s) or service(s) virtually offered by the organization, in which the homepage is processed. The web page title is proposed to represent the website as the "Application component" ArchiMate element to demonstrate the software that supports business processes of products or services delivery through the Internet. Other important ArchiMate elements "Business process" and "Application service" are proposed to be created using hyperlink "a" tags on the organization's homepage. We assume that hyperlinks reflect actions that customers can do when visiting a website to perform business activities, e.g. order a product, buy a subscription, learn a tutorial, etc. In other words, by using hyperlinks customers trigger business processes on the websites to get products or services. Using the following equation based on tuple calculus formalisms [15], a set of pairs of hyperlink text content and URL can be received:

$$HL = \{tag: \{text, attr("href")\} | tag \in TD \land tag.tagName = "a"\}, \tag{8}$$

where HL is the set of pairs of hyperlink text content and URL data.

Then, pairs of hyperlink text content and URL should be refined to detect business activities and leave behind other typical website menu items, such as “about us” or “contacts”. According to [16], verb-object activity labels represent the best practice of textual business process description. Therefore, it is proposed to refine extracted pairs of hyperlink text content and URL to keep only the verb-object “a” tags that are supposed to represent business activities. Hence, the following procedure is proposed to detect business activities in the hyperlinks present on the organization’s homepage. Process each pair of hyperlink text content and URL data $\{text, attr("href")\} \in HL$ using the following steps:

1. Remove leading and trailing whitespaces from the current hyperlink text content value $text$:

$$strip: Text \rightarrow Text_1, text_1 \in Text_1, \tag{9}$$

where $Text_1$ is the set of hyperlink text content values $text_1 \in Text_1$ with removed leading and trailing whitespaces.

2. Remove multiple whitespaces from the current hyperlink text content value $text_1$:

$$shrink: Text_1 \rightarrow Text_2, text_2 \in Text_2, \tag{10}$$

where $Text_2$ is the set of hyperlink text content values $text_2 \in Text_2$ with removed multiple whitespaces.

3. Split the current hyperlink text value $text_2$ into a bag of words:

$$split: Text_2 \rightarrow hyperlinkWords = \{hw_1, hw_2, \dots, hw_k\} \in HW, \tag{11}$$

where $hyperlinkWords$ is the bag of words of the size k that belongs to the set HW of bags each of which contains words of corresponding web page hyperlinks.

4. Tag each word in $hyperlinkWords$ as a part of a speech:

$$POS: hyperlinkWords \rightarrow hwPOS = \{\langle hw_1, pos_1\rangle, \langle hw_2, pos_2\rangle, \dots, \langle hw_k, pos_k\rangle\}, \tag{12}$$

where $hwPOS$ is the bag of tuples of the size k in which each word is accompanied by the respective part of a speech tag.

5. If the $hyperlinkWords$ bag contains at least one word and its first word hw_1 is a verb (i.e. the corresponding pos_1 is a verb in the base form or one of another verb forms [17]), then establish a mapping between hyperlink text content values and collections of URL values in the following way:

$$BA: Text_2 \rightarrow \bigcup_{text_{2i} \in Text_2} \{attr("href")_j \mid j = \overline{1, n_i}\}, \tag{13}$$

where n_i is the number of hyperlinks that correspond to the $text_{2i} \in Text_2$ text content value.

The example of using the proposed procedure (9) – (13) is demonstrated in Table 2. It shows steps 3 – 5 since the previous steps 1 – 2 are simple enough and show only trivial string transformations. The algorithm for business activities detection that formalizes steps 1 – 5 is given in Fig. 2.

Table 2

The example of the procedure of business activities detection

Step	Input	Output
3	$Text_2 = \{\{“Buy”, “/phone”\},$ $\{“Buy”, “/tablet”\},$ $\{“Manage\ account”, “/manage”\},$ $\{“About\ us”, “/about”\}\}$	$hyperlinkWords = \{\{“Buy”\},$ $\{“Buy”\},$ $\{“Manage”, “account”\},$ $\{“About”, “us”\}\}$
4	$hyperlinkWords$	$hwPOS = \{\{\langle “Buy”, “verb”\rangle\}, \quad \{\langle “Buy”, “verb”\rangle\},$ $\{\langle “Manage”, “verb”\rangle, \langle “account”, “noun”\rangle\},$ $\{\langle “About”, “preposition”\rangle, \langle “us”, “pronoun”\rangle\}\}$
5	$hwPOS$	$BA = \{(“Buy”, \{“/phone”, “/tablet”\}),$ $(“Manage\ account”, \{“/manage”\})\}$

2.2. Enterprise Architecture Landscape Construction using ArchiMate

Using the mapping between hyperlink text content values and collections of URL values (13), it is now possible to build the formal description of an ArchiMate EA model.

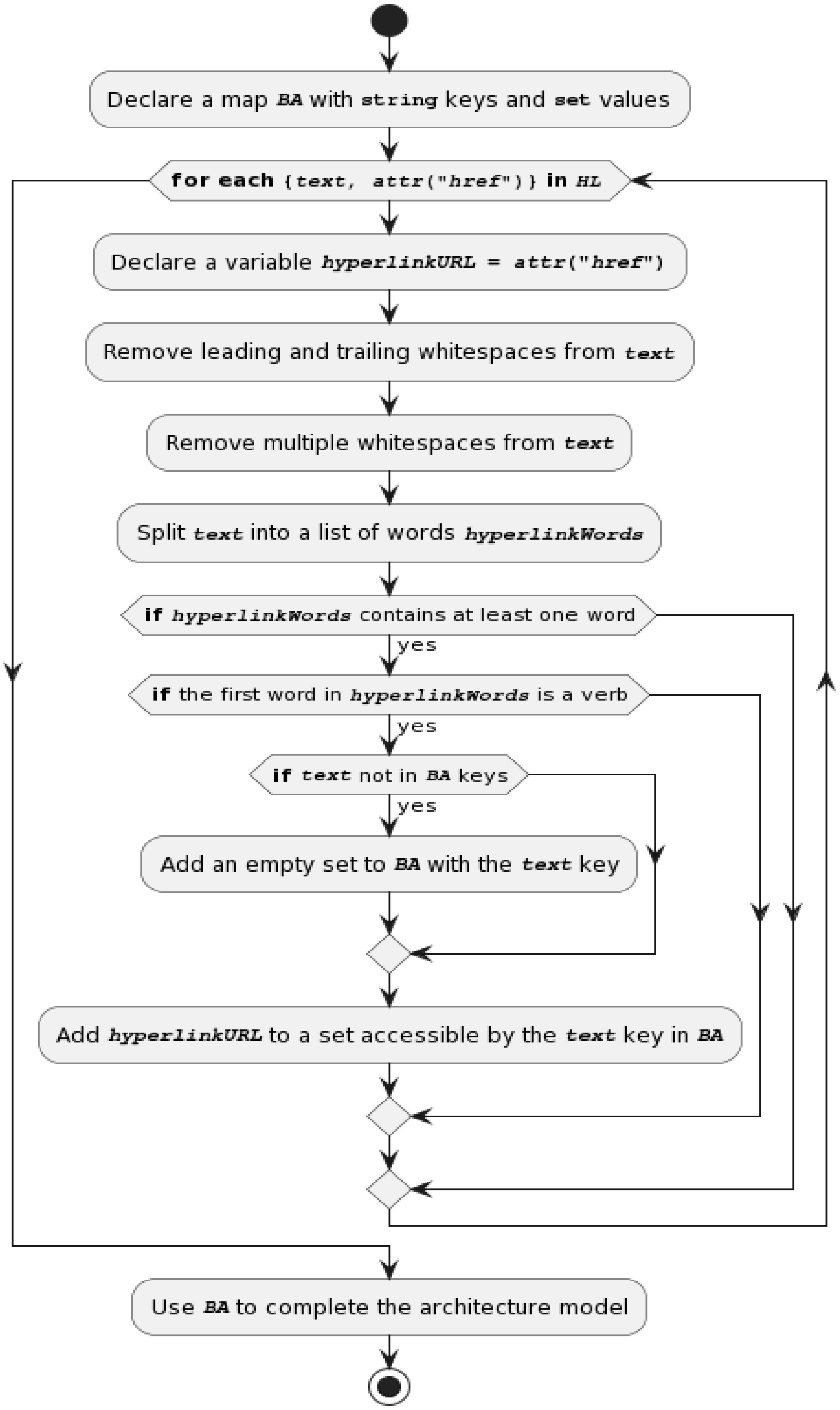

Declare a map BA with string keys and set values
for each {text, attr("href")} in HL
Declare a variable hyperlinkURL = attr("href")
Remove leading and trailing whitespaces from text
Remove multiple whitespaces from text
Split text into a list of words hyperlinkWords
if hyperlinkWords contains at least one word
yes
if the first word in hyperlinkWords is a verb
yes
if text not in BA keys
yes
Add an empty set to BA with the text key
Add hyperlinkURL to a set accessible by the text key in BA
Use BA to complete the architecture model

Figure 2: The algorithm for business activities detection

This ArchiMate EA model should include a “Business service” element based on the web page description, “Business process” elements based on hyperlink text content values, “Business service” elements based on hyperlink URL values, an “Application component” element based on the web page title, a “Technology service” element based on the web page URL, and a “Technology node” element that represents a web hosting. Relationships between EA elements mentioned above are given in Table 3 according to the syntax and semantics of the ArchiMate EA modeling language [3].

The formal description of the ArchiMate model (1) that could be constructed taking into account the suggested EA elements and relationships between them (Table 3) is given below:

$$
\begin{gathered}
AM = Build(URL, Title, Desc, BA) = \\
= \left\langle \begin{gathered}
V = \{URL, Desc, Title, "Web\ hosting"\} \cup Text_2 \cup \bigcup_{text_2 \in Text_2} BA(text_2), \\
E = \bigcup_{text_2 \in Text_2} \{text_2, Title\} \cup \bigcup_{text_2 \in Text_2} \bigcup_{attr("href") \in BA(text_2)} \{attr("href"), text_2\} \cup \\
\cup \bigcup_{text_2 \in Text_2} \bigcup_{attr("href") \in BA(text_2)} \{Title, attr("href")\} \cup \{\{URL, Title\}\} \cup \\
\cup \{\{"Web\ hosting", Title\}\}, \\
C = \{BusinessService, BusinessProcess, ApplicationService, \\
ApplicationComponent, TechnologyService, TechnologyNode\}, \\
R = \{Realization, Serving, Composition\}, \\
vt = \{(URL, TechnologyService), (Desc, BusinessService), \\
(Title, ApplicationComponent), ("Web\ hosting", TechnologyNode)\} \cup \\
\cup \bigcup_{text_2 \in Text_2} \{(text_2, BusinessProcess)\} \cup \\
\cup \bigcup_{text_2 \in Text_2} \bigcup_{attr("href") \in BA(text_2)} \{(attr("href"), ApplicationService)\}, \\
et = \{(\{URL, Title\}, Serving)\} \cup \{(\{"Web\ hosting", Title\}, Composition)\} \cup \\
\cup \bigcup_{text_2 \in Text_2} \{(\{text_2, Desc\}, Realization)\} \cup \\
\cup \bigcup_{text_2 \in Text_2} \bigcup_{attr("href") \in BA(text_2)} \{(\{attr("href"), text_2\}, Serving)\} \cup \\
\cup \bigcup_{text_2 \in Text_2} \bigcup_{attr("href") \in BA(text_2)} \{(\{Title, attr("href")\}, Realization)\}
\end{gathered} \right\rangle .
\end{gathered}
\tag{14}
$$

Table 3
Relationships between EA elements

From	To	Relationship
Technology node	Technology service	Composition
Technology service	Application component	Serving
Application component	Application service	Realization
Application service	Business process	Serving
Business process	Business service	Realization

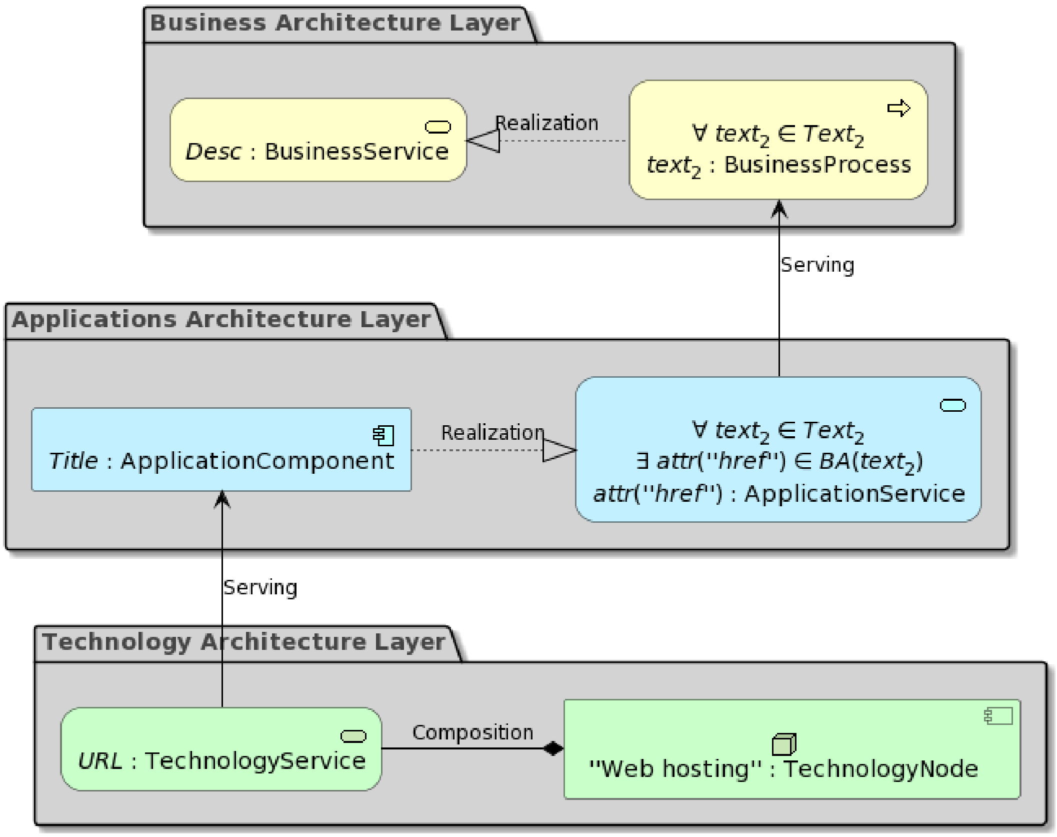
The proposed formal description of the EA model (14) can be used to build the respective ArchiMate model according to the following metamodel (Fig. 3). The proposed metamodel (Fig. 3) demonstrates the EA landscape that could be obtained from organizational web page processing. As is given in (14) and in Fig. 3, the EA landscape focuses more on business processes and supportive application services. In contrast, other EA elements, such as business services, application components, and technology architectural elements are used to complete the cross-layer nature of the ArchiMate language.

2.3. Software Implementation of the EA Web Mining Approach

The proposed "EA web mining" approach was implemented as the software tool used to extract EA landscapes from corporate websites. The software tool was created using the Python programming language thanks to its relative simplicity but meanwhile powerful packages to work with HyperText Transfer Protocol (HTTP) requests, HTML pages, regular expressions, and natural language.

The software implementation includes the main module serving as the application's endpoint. It depends on four modules corresponding to the proposed approach's steps (Fig. 1). These are the following software modules:

- "Web Page Parsing" – this module is responsible for HTML page parsing to work with tags, attributes, and text contents;
- "Data Extraction" – this module is responsible for the title and description tags processing, as well as URL address and text content data extraction from web page hyperlinks;
- "Business Activities Detection" – this module is responsible for hyperlinks processing to detect the ones that mean certain business activities that trigger business processes supported by the web application services;
- "EA Generation" – this module is responsible for ArchiMate model generation using EA elements and relationships formulated on the previous steps and formally described by (14); the output files are produced in the Plant UML diagramming language [18].



Business Architecture Layer
Desc : BusinessService
Realization
$\forall\, text_2 \in Text_2$
$text_2$: BusinessProcess
Serving
Applications Architecture Layer
Title : ApplicationComponent
Realization
$\forall\, text_2 \in Text_2$
$\exists\, attr(\text{"}href\text{"}) \in BA(text_2)$
$attr(\text{"}href\text{"})$: ApplicationService
Serving
Technology Architecture Layer
URL : TechnologyService
Composition
"Web hosting" : TechnologyNode

Figure 3: The EA landscape metamodel

The software structure is given in a component diagram below (Fig. 4). According to the demonstrated above software component diagram (Fig. 4), the third-party Python modules are also used by the application. There are the following modules in use:

- "urllib.request" – this module helps to make HTTP requests and open URLs taking into account the authentication, redirections, cookies, and other features [19];
- "bs4" or "Beautiful Soup" – this module helps to pull data out of HTML and eXtensible Markup Language (XML) files [20];
- "re" – this module provides regular expression operations [21];
- "nltk" or "Natural Language Toolkit" – this module helps to work with human language data in Python [22].

Hence, the "urllib.request" module is used by the created software tool to parse web pages, the "bs4" module is used to extract data from HTML pages, while "re" and "nltk" modules are used to process extracted data from web pages and detect possible business activities offered by corporate homepages. The "Natural Language Toolkit" module plays a core role in the implemented algorithm for business activity detection (Fig. 2). It is used for the part of speech tagging of hyperlink text content words to detect the hyperlinks that begin with verbs. Then, according to the verb-object activity labeling best practice [16], such hyperlinks are used as sources for business process and application service elements extraction according to the suggested algorithm (Fig. 2).

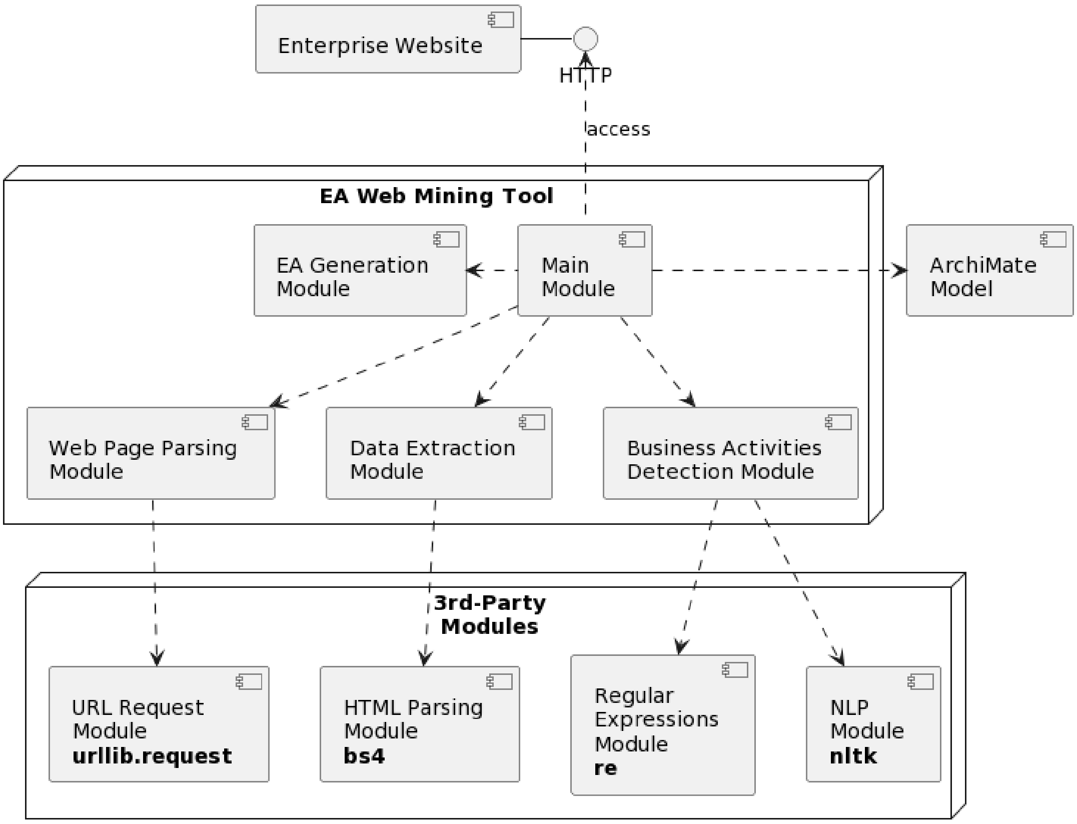


Figure 4: The software component diagram

3. Results and Discussion

To demonstrate the capabilities of the proposed “EA web mining” approach and the corresponding software tool (Fig. 4), let us select for processing websites of two well-known enterprises that belong to the telecommunications industry. As the result, we expect to obtain EA models revealing business processes that could be triggered by users of these websites to receive services or order products.

The first telecommunications enterprise whose website we used as the source for “EA web mining” is T-Mobile (Fig. 5) [23]. A closer look at the extracted business processes is given in Fig. 6. This model demonstrates only the business process architecture, while other EA elements and relationships (Fig. 5) are avoided. Another telecommunications enterprise whose website we used as the source for “EA web mining” is Verizon (Fig. 7) [24]. A closer look at the extracted business processes is given in Fig. 8. This model demonstrates only the business process architecture, while other EA elements and relationships (Fig. 7) are avoided. For the sake of EA models’ readability, the names of business services in Fig. 5 and Fig. 7 were changed to “…” because the respective hyperlink URLs could be of significant length and, therefore, may horizontally overflow the models by making them unclear for a reader. Thus, automatically designed T-Mobile (Fig. 5) and Verizon (Fig. 7) EA models contain 16 (Fig. 6) and 12 (Fig. 8) business processes respectively. However, there are “false positive” business processes that do not correspond to the verb-object activity labeling style [16]:

- “Unlimited Phone Plans” and “Unlimited Age 55+” elements of the T-Mobile EA model;
- “Certified pre-owned phones”, “Certified pre-owned watches”, “Charging”, “Gaming”, “Unlimited”, “Connected devices”, “Connected car plans”, and “Moving” elements of the Verizon EA model.

Therefore, we can introduce the following quality measures:

- TP is the number of “true positive” detected business processes – 14 for the T-Mobile EA model (Fig. 6) and 4 for the Verizon EA model (Fig. 8);
- FP is the number of “false positive” detected business processes – 2 for the T-Mobile EA model (Fig. 6) and 8 for the Verizon EA model (Fig. 8).

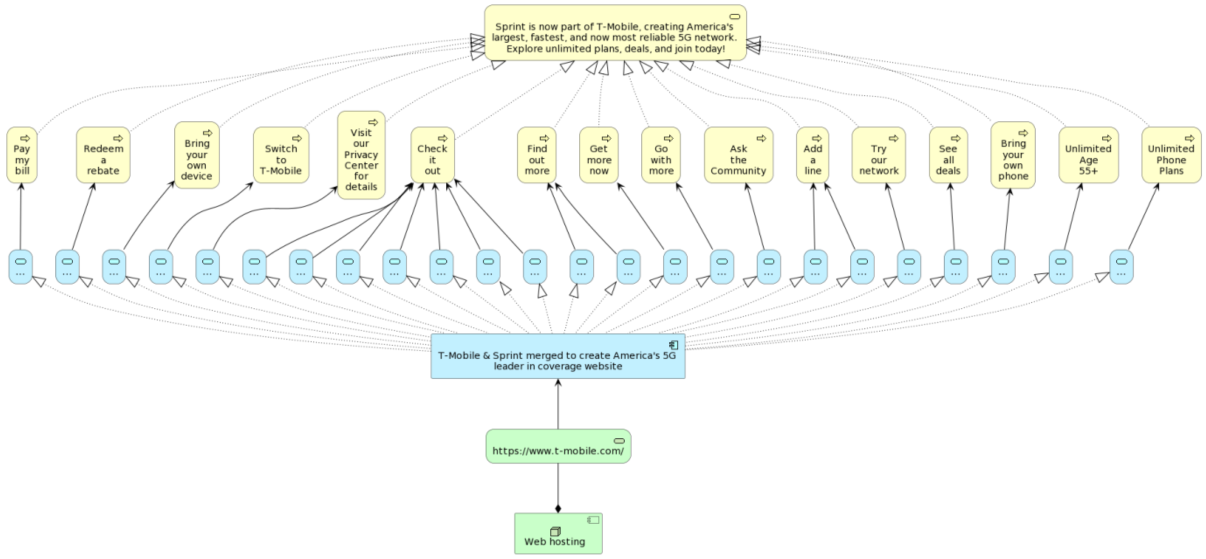

Sprint is now part of T-Mobile, creating America's largest, fastest, and now most reliable 5G network. Explore unlimited plans, deals, and join today!
Pay my bill
Redeem a rebate
Bring your own device
Switch to T-Mobile
Visit our Privacy Center for details
Check it out
Find out more
Get more now
Go with more
Ask the Community
Add a line
Try our network
See all deals
Bring your own phone
Unlimited Age 55+
Unlimited Phone Plans
...
T-Mobile & Sprint merged to create America's 5G leader in coverage website
https://www.t-mobile.com/
Web hosting

Figure 5: The EA model created as the result of T-Mobile homepage processing

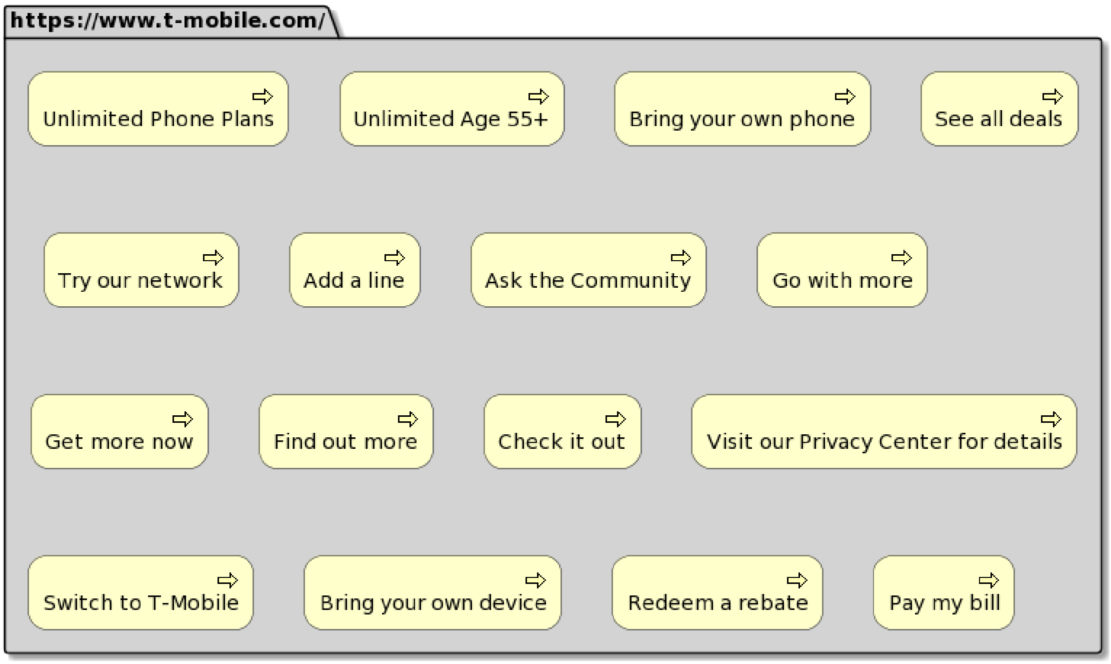


Figure 6: The business process architecture created as the result of T-Mobile homepage processing

Hence, the precision of the proposed “EA web mining” approach could be measured as follows:

$$Precision = \frac{TP}{TP + FP} = \frac{14 + 4}{(14 + 4) + (2 + 8)} = \frac{18}{28} = 0.64. \qquad (15)$$

The calculated precision measure (15) signalizes that 64% of detected business process elements are representing business activities offered by the considered websites [23] and [24]. The remaining elements recognized as “business processes” are representing offers that inform customers but do not usually require any active behavior, such as “bring”, “pay”, “report”, “try”, etc. Such elements could be changed from active to passive ArchiMate structure elements, such as “business objects”.

The precision measure could be improved by introducing more advanced methods and techniques for business activity detection, i.e. using neural networks or other machine learning facilities.

However, the final decision on EA design, including possible adjustments, must be made by the EA model designer, since the final goal of automatic EA modeling is to reduce the time and cost consumption of enterprise architecture modeling, while keeping models accurate and relevant to a modeling domain.

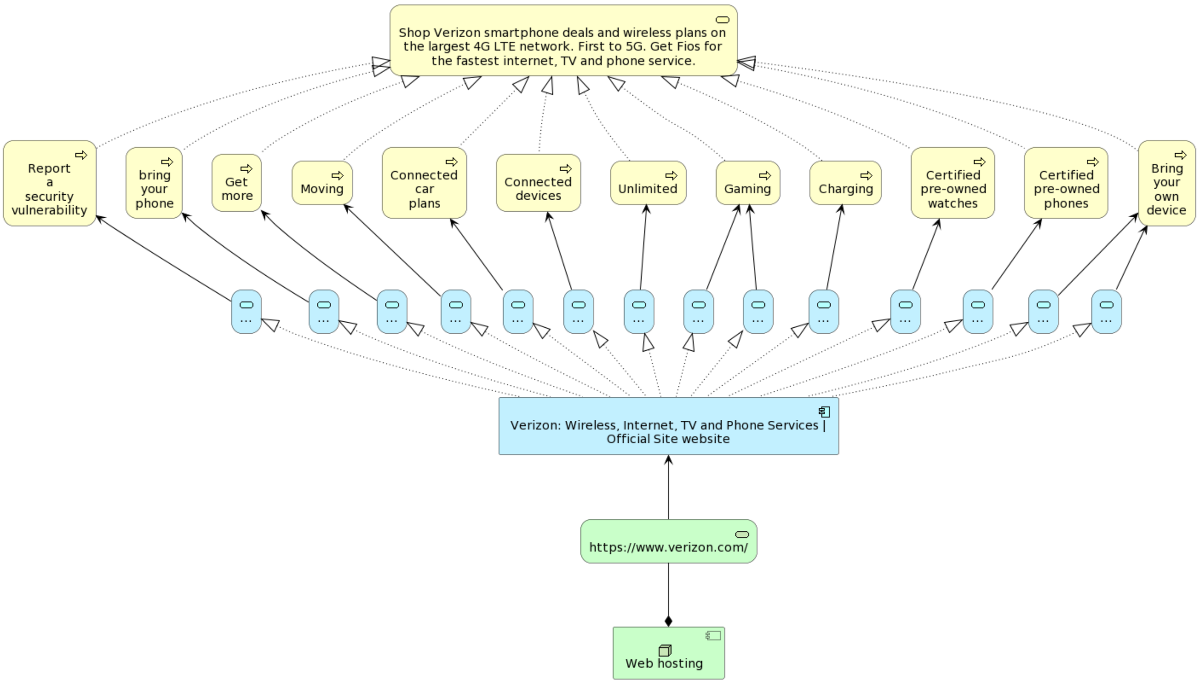

Shop Verizon smartphone deals and wireless plans on the largest 4G LTE network. First to 5G. Get Fios for the fastest internet, TV and phone service.
Report a security vulnerability
bring your phone
Get more
Moving
Connected car plans
Connected devices
Unlimited
Gaming
Charging
Certified pre-owned watches
Certified pre-owned phones
Bring your own device
...
Verizon: Wireless, Internet, TV and Phone Services | Official Site website
https://www.verizon.com/
Web hosting

Figure 7: The EA model created as the result of Verizon homepage processing

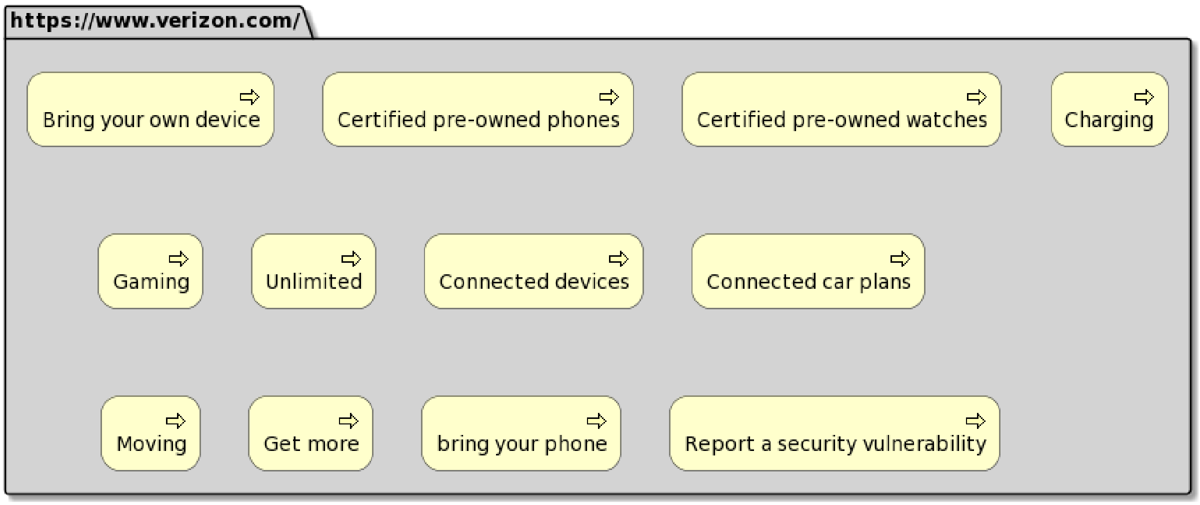


Figure 8: The business process architecture created as the result of Verizon homepage processing

4. Conclusion and Future Work

In this paper, we proposed the approach and the software tool for the automatic building of EA models using corporate websites. The proposed technique is named “enterprise architecture web mining” and aims to simplification of the process of enterprise architecture blueprinting in the early stages of EA development. It is expected that the proposed approach can reduce the time and cost consumption of EA modeling by making it possible to construct business process-centric EA landscapes directly from company homepages. The proposed approach uses HTML parsing techniques to extract data from enterprise web pages. It considers “title” and “description” meta tags as the sources of general business information, and hyperlink tags as the sources of business activity information. Hyperlink text content values are checked for matching the verb-object labeling style for the sake of business activity recognition among all web page hyperlinks. Then detected business activities are represented as ArchiMate business processes together with remaining EA elements, such as the business service (based on the web page description), application services (based on the hyperlink URL values), the application component (based on the web page title), the technology service (based on the web page URL), and the technology node (it represents a web hosting). The software implementation of the proposed approach is based on the Python language with its modules for HTTP request handling, HTML file parsing, regular expression matching, and natural language processing. The software tool was used to apply the “EA mining” technique to build EA models based on T-Mobile and Verizon homepages. Obtained ArchiMate EA models demonstrate business processes discovered on these web pages and the supporting EA elements and relationships. Additional business process architecture models were also built and analyzed taking into account the precision measure. Obtained EA models and their analysis results demonstrate the 64% precision of the suggested “EA mining” technique. Future work in this field should include the elaboration of business activity detection in enterprise web pages.

5. References

[1] A. Josey et al., TOGAF® Business Architecture Level 1 Study Guide, TOGAF series, Van Haren, 2019.

[2] Y. Masuda, M. Viswanathan, Enterprise Architecture for Global Companies in a Digital IT Era: Adaptive Integrated Digital Architecture Framework (AIDAF), Springer, 2019.

[3] A. Josey, ArchiMate® 3.0.1 – A Pocket Guide, Van Haren, 2017.

[4] J. D. McDowall, Complex Enterprise Architecture: A New Adaptive Systems Approach, Apress, 2019.

[5] V. Kale, Digital Transformation of Enterprise Architecture, CRC Press, 2019.

[6] A. Zimmermann, R. Schmidt, L. C. Jain, Architecting the Digital Transformation: Digital Business, Technology, Decision Support, Management, Springer Nature, 2020.

[7] H. A. H. Handley, The Human Viewpoint for System Architectures, Springer, 2019.

[8] A. Buchalcevova, Software process improvement in small companies as a path to enterprise architecture, Information Systems Development, Springer, New York, NY, 2013, pp. 243–253. doi:10.1007/978-1-4614-4951-5_20.

[9] E. Moyle, D. Kelley, Practical Cybersecurity Architecture: A guide to creating and implementing robust designs for cybersecurity architects, Packt Publishing Ltd, 2020.

[10] A. Fleischmann, S. Oppl, W. Schmidt, C. Stary, Contextual Process Digitalization: Changing Perspectives – Design Thinking – Value-Led Design, Springer Nature, 2020.

[11] A. Fajri, Enterprise Architecture Mining, MS thesis, University of Twente, 2019.

[12] R. Pérez-Castillo, F. Ruiz, M. Piattini, A decision-making support system for Enterprise Architecture Modelling, Decision Support Systems 131 (2020) 113249. doi:10.1016/j.dss.2020.113249.

[13] R. Perez-Castillo et al., A systematic mapping study on enterprise architecture mining, Enterprise Information Systems 5(13) (2019) 675–718. doi:10.1080/17517575.2019.1590859.

[14] D. Orlovskyi, A. Kopp, Enterprise Architecture Modeling Support based on Data Extraction from Business Process Models, CEUR Workshop Proceedings 2608 (2020) 499–513. URL: http://ceur-ws.org/Vol-2608/paper38.pdf.

[15] S. W. Dietrich, Understanding Databases: Concepts and Practice, John Wiley & Sons, 2021.

[16] J. Mendling, Managing structural and textual quality of business process models, International Symposium on Data-Driven Process Discovery and Analysis, Springer, Berlin, Heidelberg, 2012, pp. 100–111. doi:10.1007/978-3-642-40919-6_6.

[17] Categorizing and Tagging Words. URL: https://www.nltk.org/book/ch05.html.

[18] Plant UML. URL: https://plantuml.com/.

[19] urllib.request – Extensible library for opening URLs. URL: https://docs.python.org/3/library/urllib.request.html.

[20] Beautiful Soup Documentation. URL: https://www.crummy.com/software/BeautifulSoup/bs4/doc/.

[21] re – Regular expression operations. URL: https://docs.python.org/3/library/re.html.

[22] Natural Language Toolkit. URL: https://www.nltk.org/.

[23] T-Mobile. URL: https://www.t-mobile.com/.

[24] Verizon. URL: https://www.verizon.com/.