Preprint version of the published paper

Buchalcevova, A. (2019). Using ArchiMate to model ISO/IEC 29110 standard for very small entities. *Computer Standards & Interfaces*, *65*, 103-121.

https://doi.org/10.1016/j.csi.2019.03.001

accessible at

https://www.sciencedirect.com/science/article/abs/pii/S0920548918303386

Using ArchiMate to Model ISO/IEC 29110 Standard for Very Small Entities

Alena Buchalcevova
Department of Information Technologies University of Economics, W. Churchill sqr 4, 13067 Prague, Czech Republic

Keywords:
ISO/IEC 29110, ArchiMate, Mapping, Modeling, Basic Profile

Abstract
Paper concentrates on the ISO/IEC 29110 standard aimed at improving systems and software engineering life cycle processes in very small entities, which has been recently developed and started to be implemented worldwide. With the purpose to facilitate development and maintenance of this standard and its implementation in very small entities utilization of ArchiMate is proposed. The ArchiMate is a widely accepted open standard for modeling Enterprise Architecture supported by a variety of modeling tools, which was also successfully used beyond Enterprise Architecture domain. The contribution of this paper lies in (1) the development of the ISO/ IEC 29110 Basic Profile Metamodel; (2) the definition of the mapping between the ISO/IEC 29110 Basic Profile Metamodel and ArchiMate; (3) its evaluation using the Bunge-Wand-Weber (BWW) model; (4) the application of the mapping resulting in the development of the ISO/IEC 29110 Basic Profile ArchiMate Model being implemented in a modeling tool that is freely available and can be used by VSEs, assessors and standard developers.

1. Introduction

Recently, the ISO/IEC 29110 systems and software engineering standard aimed at improving life cycle processes in Very Small Entities (VSEs) has been developed and started to be implemented

worldwide, mainly in Canada, Mexico and Thailand [1]. Experience gained from the pilot projects and ISO/IEC 29110 standard implementations has been concluded and published [2–4] showing that the implementation has been predominantly successful, however with certain issues arising. Lessons learned from a review of 9 case studies of ISO/IEC 29110 implementations in a Canadian context [2] pointed out the issues connected with a customization of the standard to agile practices used in VSEs, which was also stated in [3]. Further, an issue of customization of the standard terminology to a terminology used by VSEs was identified along with the need for a support of the transition from ad-hoc processes to defined processes [2]. The implementation of the ISO/IEC 29110 standard in an IT startup in Peru described in [3] emphasized the challenges connected with the translation of deployment packages that support the implementation of the standard to Spanish and the need of the identification of relationships between work products and objectives they contribute to achieve. Further requirements for a ISO/IEC 29110 standard improvement were stated in [4] as for example (1) provide detailed guidelines and assistance; (2) align with existing company business and development processes; (3) align with other specific software technical standards and processes.

The above mentioned issues are connected with the fact that software and system development is a complex activity [5], which is highly sensitive to human interaction and team work [6]. Understanding that modeling represents an essential tool for coping with complexity and enables to communicate ideas among team members and different stakeholders efficiently, the motivation for this research is thus to use the ArchiMate modeling language to express the content of the ISO/IEC 29110 Basic Profile and this way to facilitate its implementation in Very Small Entities. More specifically, the purpose of this paper is as follows.

> **Purpose**: to define specific mapping between the ISO/IEC 29110 Basic Profile and ArchiMate, which enables to develop the ISO/IEC 29110 Basic Profile ArchiMate Model aimed at improving Basic Profile implementation in Very Small Entities.

To confirm the applicability of the selected modeling language, ArchiMate is a widely accepted open standard for modeling Enterprise Architecture that has also been used in other domains (see Section 2.3). It has a large user base and a variety of supporting modeling tools [7].

The ISO/IEC 29110 standard currently uses visualization just rarely and only in the form of unformal images as shown in Figures 11 and 13 embodied in Section 7. Using the standard modeling notation should therefore improve the quality of the standard and enable VSEs to effectively customize the standard as well as all supporting materials such as deployment packages by using widely accessible modeling tools.

The contribution of this paper lies in (1) the development of the ISO/IEC 29110 Basic Profile Metamodel; (2) the definition of the mapping between the ISO/IEC 29110 Basic Profile Metamodel

and ArchiMate; (3) its evaluation using the Bunge-Wand-Weber (BWW) model; (4) the application of the mapping resulting in the development of the ISO/IEC 29110 Basic Profile ArchiMate Model being implemented in a modeling tool that is freely available and can be used by VSEs, assessors and standard developers. To endorse the contribution, the Basic Profile ArchiMate Model has been evaluated using expert interviews.

The paper is structured as follows. Following the Introduction, Section 2 provides the research background, i.e. introduces both the ISO/IEC 29110 standard and ArchiMate modeling language along with the ArchiMate usage. In Section 3, the methodology of the research is described. Section 4 then describes the developed ISO/IEC 29110 Basic Profile Metamodel. Further, the mapping of the ISO/IEC 29110 Basic Profile to the ArchiMate language is introduced in Section 5 and evaluated using the Bunge-Wand-Weber (BWW) model in Section 6. In Section 7, the proposed mapping is applied and the ISO/IEC 29110 Basic Profile ArchiMate Model is developed. The model evaluation using expert interviews is described in Section 8. In the Discussion Section, the results are discussed along with the research limitations and recommendations for further research. Lastly, overall contribution and concluding remarks are stated.

2. Research Background

2.1 ISO/IEC 29110 Standard

Although very small companies with their core business in software development have a significant influence on the economy, most of them do not implement any international standards or models like the ISO/IEC 12207 or CMMI [8,9]. Subsequently, these companies have only very limited opportunities to be recognized as entities that produce quality software, and thus are often cut off from prospective contracts. In order to help small companies to improve their software processes and be recognized as entities that produce quality software, the ISO/IEC 29110 standard has been developed. The term “Very Small Entity” (VSE) was defined by the ISO/IEC JTC1/SC7 Working Group 24 and consequently adopted in the emerging ISO/IEC 29110 standard meaning “an entity (enterprise, organization, department or project) that has up to 25 people” [10].

The ISO/IEC 29110 standard’s structure is as follows. Main concepts, terms and structure of the standard are explained in Part 1 Overview [10]. Part 2 Framework and Taxonomy [11] presents the principles and mechanism of building the VSE Profiles that represent a key concept of the ISO/IEC 29110 standard. Part 3 then defines the process assessment guidelines and compliance requirements needed to meet the objectives of defined VSE Profiles. This part of the standard is used by certified assessors to perform a VSE assessment. It comprises four standards, i.e. Assessment Guide [12], Conformity Certification Scheme [13], Certification Requirements for Conformity Assessments of VSE Profiles using Process Assessment and Maturity Models [14] and Autonomy-based Improvement

Method [15]. Part 4 Specifications of VSE Profiles [16] provides a mapping to the source standards, e.g. the ISO/IEC 12207, and is very useful for method developers and assessors On the contrary, Part 5 is intended for VSEs themselves and consists of several Management and Engineering Guides designated for each Profile as described in the next paragraph.

The Working Group 24 started with developing VSE Profiles in the field of software engineering. The "Generic" Profile Group was defined being applicable to a vast majority of VSEs that do not develop critical software. Within this group, four VSE Profiles were proposed, i.e. Entry, Basic, Intermediate, and Advanced. By using these Profiles, very small companies have the chance to improve their processes in a clear and stepwise manner. The Basic Profile for software engineering [17] intended for a single project with no special risks or situational factors was developed and published first. As particular pilot projects of the Basic Profile implementation in VSEs showed, this Profile was still quite difficult to be implemented by some companies. For this reason, the Entry Profile [18] was developed as a tool for simplification that applies to a small project (a six person-months effort) or start-up VSE. The Intermediate Profile [19] is meant for a VSE which handles more than one project at a time, and therefore is aware of assigning project resources and monitoring projects to accomplish business objectives and customer satisfaction. Lastly, the Advanced Profile [20] describes processes targeted at VSE that wants to sustain and grow as an independent competitive software development business. Beyond these four profiles, the Organizational Management Guidelines [21] were developed and published focused on Organizational Management, Project Portfolio Management, Resource Management and Process Management Processes.

Following the development of software engineering profiles, the Working Group 24 expanded its focus also on the area of systems engineering as described in [22]. At present, the Entry Profile [23] and the Basic Profile [24] for systems engineering are published. Recent effort in the ISO/IEC 29110 standard development is aimed at incorporating agile practices into the standard based on the analysis published in [25] and developing profiles for delivery of services, where the Service Delivery Guidelines [26] has already been published. The entire history of the ISO/IEC 29110 standard development as well as actual status of its implementation is outlined in [1,27].

To help VSEs with a Profile implementation, a series of Deployment Packages were developed and offered free of charge [28]. A Deployment Package acts as a detailed methodology that guides a company through the process of the Profile implementation. A typical Deployment Package includes process descriptions, activities, tasks, roles and products, templates, checklists, examples, references and mapping to the standards and models, and a list of supporting tools.

Conducting pilot projects is then recommended serving as efficient means to accelerate the adoption and utilization of the ISO/IEC 29110 standard by VSEs. Pilot projects can reduce risks and enable recognizing organizational and technical issues associated with the deployment of new software

engineering practices [27]. To further assist with the roll out of a pilot project and to ensure that all pilot projects are conducted similarly around the world, a set of pilot project guidelines were developed in the form of a Deployment Package to describe the process of conducting pilot projects.

2.2 ArchiMate

ArchiMate is a relatively new (version 1.0 published in 2009) Enterprise Architecture modeling language. The ArchiMate Specification is an Open Group Standard, that is supported by different tool vendors and consulting firms. ArchiMate was developed to solve the issues identified among various until then existing IT and business modeling languages such as poorly defined relations between domains, lack of model integration and clearly defined semantics, weak formal basis and lack of an overall architecture vision [29]. ArchiMate offers an integrated architectural approach that describes and visualizes the different architecture domains and their underlying relations and dependencies [30]. Thus, it allows to describe the construction and operation of business processes, organizational structures, information flows, IT systems, technical infrastructure, motivation and strategy elements. ArchiMate elements hold the middle among the detailed concepts, which are used for modeling individual domains like the Unified Modeling Language (UML) used for modeling software products and Business Process Modeling and Notation (BPMN) used for business process modeling.

The structure of ArchiMate in the latest version 3.0.1 is depicted in Figure 1. The ArchiMate elements are organized horizontally into the Layers and vertically into the Aspects. Three basic Layers, i.e. Business Layer, Application Layer and Technology Layer, represent the so called ArchiMate Core Framework. The ArchiMate Business Layer provides concepts such as Business Actor, Role, Collaboration, Artifact, Interface, Process, Service and Event concepts to model the business architecture domain. The ArchiMate Application Layer provides concepts such as Application Component, Collaboration, Interface, Function, Interaction, Service and Data Object to model the application architecture. The ArchiMate Technology Layer then provides the concepts such as Infrastructure Node, Device, System Software, Network, Communication Path, Function and Service concepts to support modeling the infrastructure layer [30]. In addition to the three main architecture layers, ArchiMate has also provided two extensions since version 2.0: Motivational concepts, and Implementation and Migration concepts, that support modeling the Enterprise Architecture implementation such as Work Packages, Deliverables, Plateau and Gap. In 2017, the latest ArchiMate 3.0.1 Specification [31] was published. New elements for modeling the organization at a strategic level were added, such as Capability, Resource, and Course of Action as well as elements modeling physical world of materials and equipment. Furthermore, the consistency and structure of the language were improved, definitions aligned with other standards, and its usability enhanced in various other ways [30]. The full ArchiMate Framework Version 3.0.1 is depicted in Figure 1.

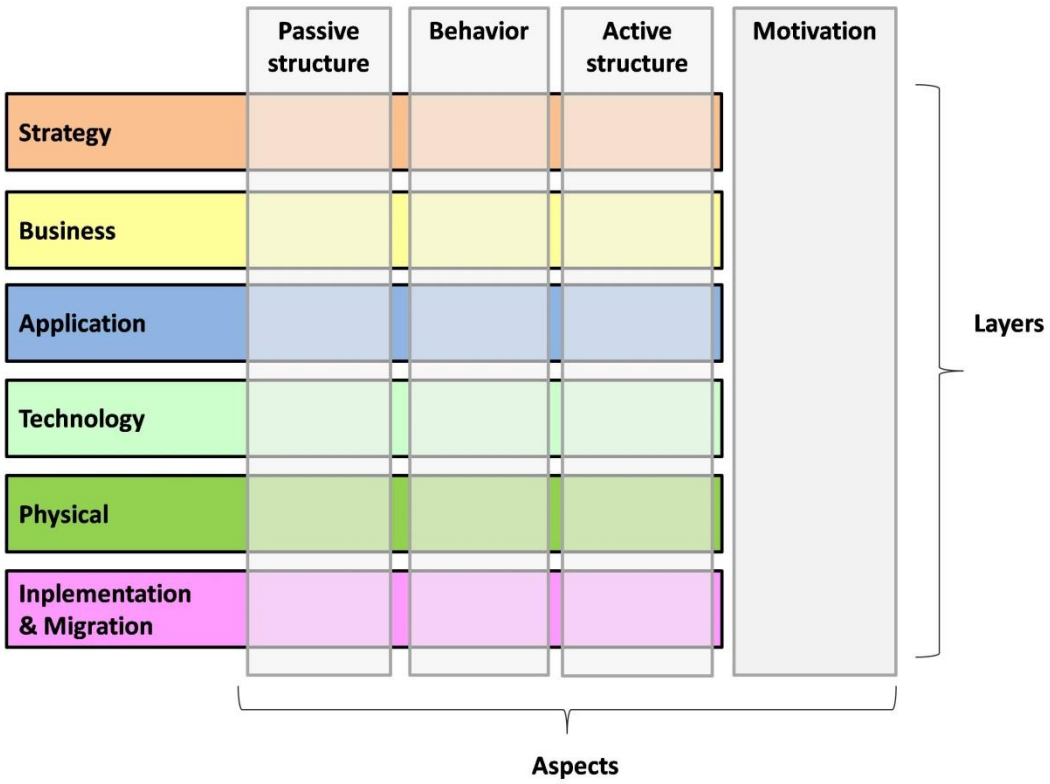


Fig. 1. Full ArchiMate Framework based on ArchiMate website [31]

Besides the layered structure, ArchiMate also distinguishes between the Active Structure Elements, Behavior Elements and Passive Structure Elements. These three aspects were inspired by natural language, where a sentence has a subject (active structure), a verb (behavior), and an object (passive structure).

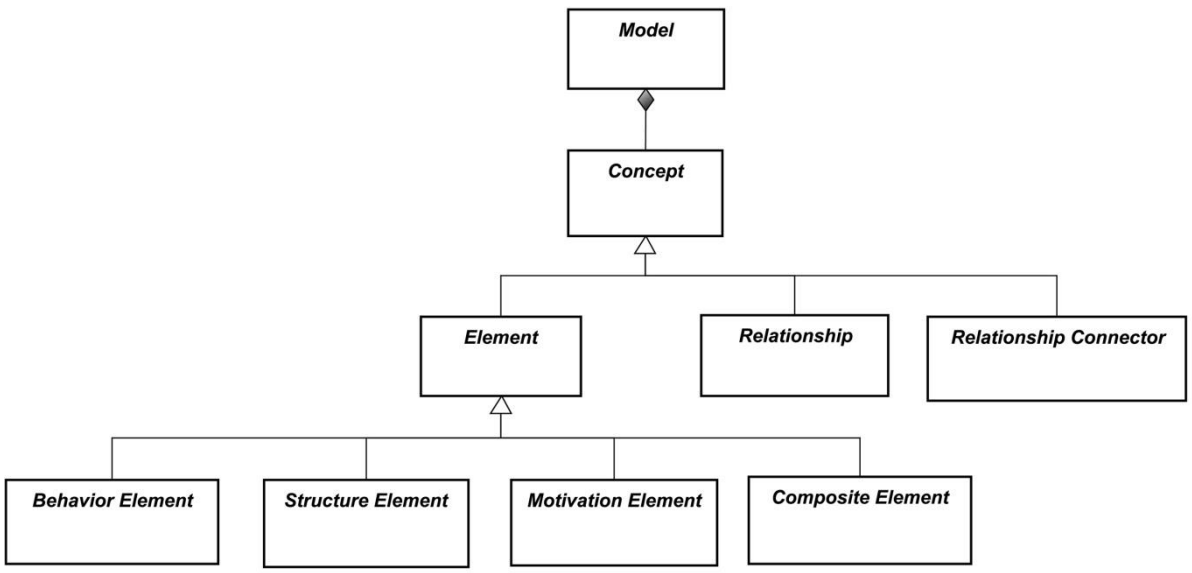


Fig. 2. Top-Level Hierarchy of ArchiMate Concepts based on ArchiMate website [31]

Figure 2 depicts top-level hierarchy of ArchiMate concepts modeled in the UML Class diagram. A Model is a collection of Concepts. A Concept is specialized into Element, Relationship and Relationship Connector. An Element is further specialized into Behavior, Structure, Motivation and Composite Elements. The Behavior Element describes a particular activity aspect, e.g. a Business Process. The Structure Element is further classified as active and passive. The Active Structure Element is capable of initiating and performing behavior. An example of an Active Structure Element is a Business Role. The Passive Structure Element is a receiver of behavior performed on the element itself. An example of a Passive Structure Element is a Business Object. The Motivation Element captures the motives requiring an Enterprise Architecture to take on a particular form and function for example a Goal. The Composite Element then groups other elements in a useful form as for example a Location.

As all classes in Figure 2 are abstract classes, their names are in italics. The most important elements are the structure and behavior elements. Their detailed hierarchy is shown in Figure 3. Concrete classes that represent ArchiMate elements are depicted in grey. Abstract classes on the lowest level like Process, Service, Interface and other are then specialized according to layers. For example, the abstract class Process has specializations such as Business Process, Application Process and Technology Process.

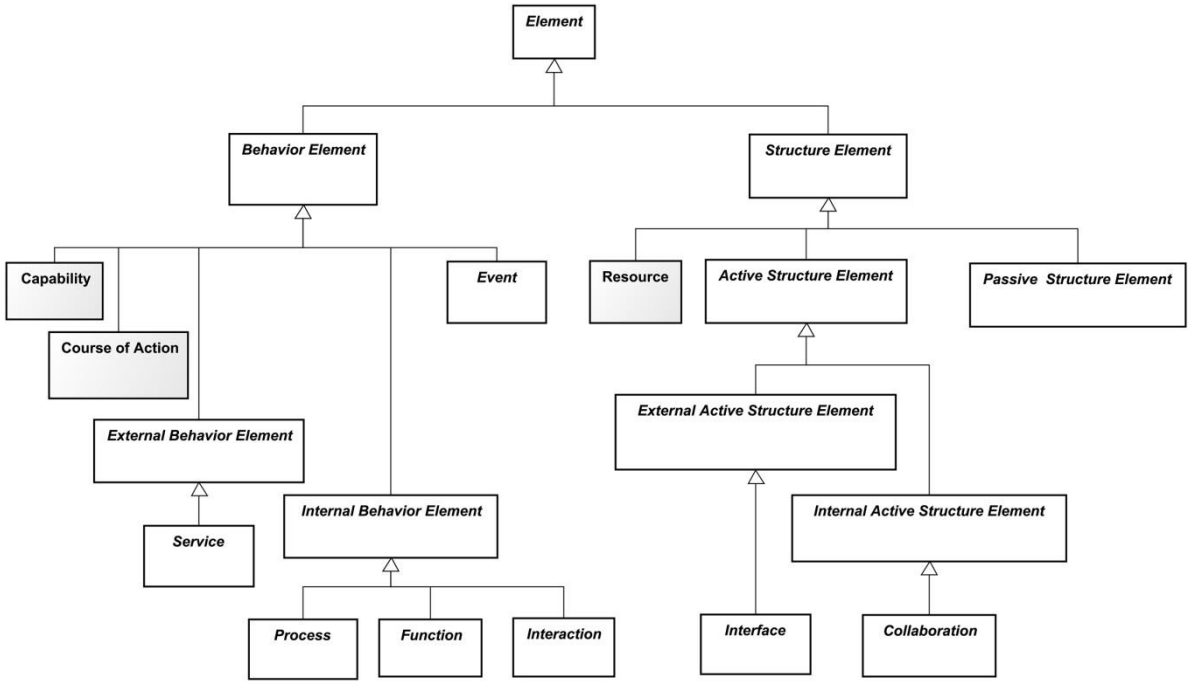


Fig. 3. Hierarchy of Behavior and Structure Elements based on ArchiMate website [31]

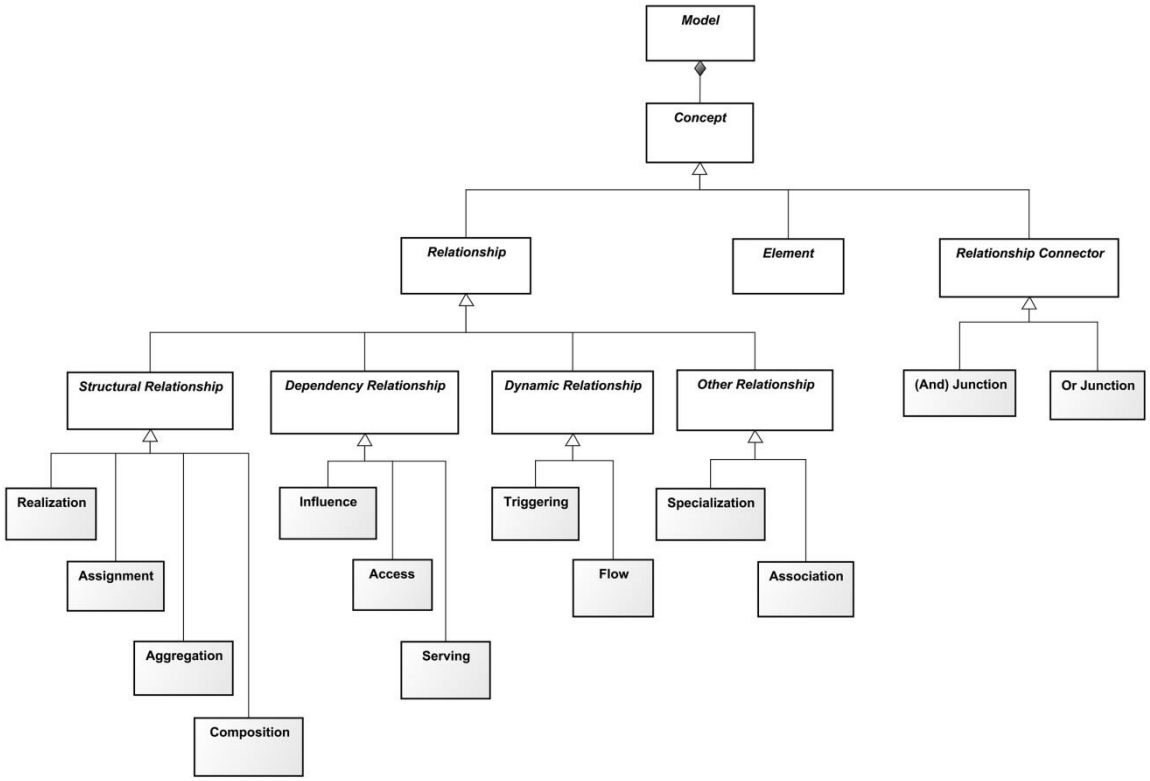


Fig. 4. Overview of Relationships based on ArchiMate website [31]

The ArchiMate language offers a rich set of relationships that represent the context in which one element is associated with another. Figure 4 shows the categorization of relationships and all relationships available (grey classes). Definitions of these relationships as well as the ArchiMate notation are presented in Table 1 [31].

Table 1 ArchiMate Relationships

Structural Relationships		**Notation**
Composition	Indicates that an element consists of one or more other concepts.	
Aggregation	Indicates that an element groups a number of other concepts.	
Assignment	Expresses the allocation of responsibility, performance of behavior, or execution.	
Realization	Indicates that an entity plays a critical role in the creation, achievement, sustenance, or operation of a more abstract entity.	
Dependency Relationships		**Notation**
Serving	Models that an element provides its functionality to another element.	
Access	Models the ability of behavior and active structure elements to observe or act upon passive structure elements.	
Influence	Models that an element affects the implementation or achievement of a particular motivation element.	+/-
Dynamic Relationships		**Notation**
Triggering	Describes a temporal or causal relationship between elements.	
Flow	Transfer from one element to another.	

Other Relationships		Notation
Specialization	Indicates that an element is a particular kind of another element.	
Association	Models an unspecified relationship, or a relationship that is not represented by another ArchiMate relationship.	
Junction	Used to connect relationships of the same type.	(And) Junction Or Junction

ArchiMate also specifies which relationships are permitted between elements. This is defined in the metamodel. Part of this metamodel that specifies main relationships between the behavior and structure elements is shown in Figure 5. The name of a relationship signifies the role of the source element in the relationship; e.g., a service serves an internal behavior element. This figure does not show all permitted relationships, every element in the language can also have composition, aggregation, and specialization relationships to elements of the same type. Furthermore, there are indirect relationships that can be derived [31].

The ArchiMate modeling language is service-oriented which means that layers are connected by a ”service orientation” paradigm, where each layer exposes functionality in the form of a service to the layer above. The use of services supports flexibility and adaptability to change as it reduces the direct dependencies that might otherwise exist in the architecture. The consumer and provider of the service only interact through a service interface, which one could think of as a channel for accessing the service [32] .

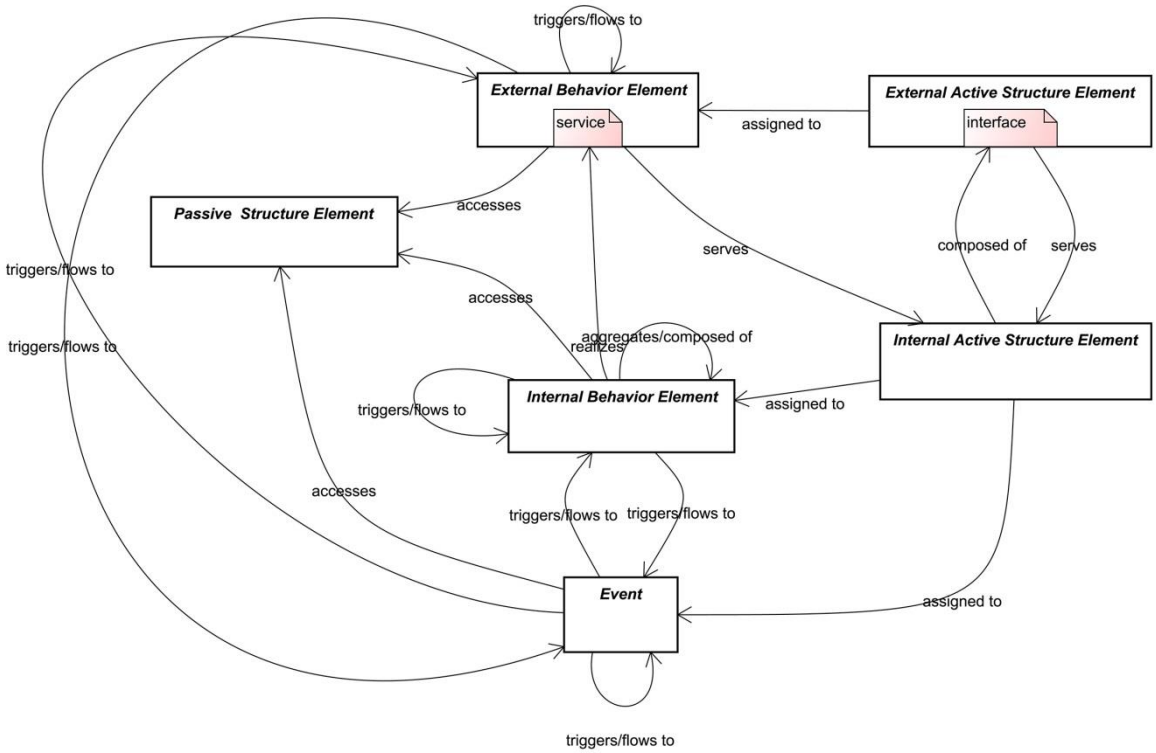


Fig. 5. Behavior and Structure Elements Metamodel based on ArchiMate website [31]

In the ArchiMate language, a Service (whether Business, Application, or Technology) is seen as an External Behavior Element and the Interface assigned to the Service is an External Active Structure Element. A Service is delivered by an Internal Behavior Element, such as a Business Process in the case of a Business Service. For the service delivery the ArchiMate relationship "realized by" is used. An Internal Active Structure Element is assigned to the internal behavior, to represent the entity that performs this behavior, behind the service interface.

The ArchiMate specification [31] does not prescribe the use of color in models. However, in a number of example models presented in the standard, colors are used to distinguish between the layers of the ArchiMate Core Framework as follows: Yellow for the Business Layer, Blue for the Application Layer and Green for the Technology Layer. The standard notation also uses a convention related to the shape of the corners of its symbols for different element types as follows: Square corners are used to denote the structure elements, Round corners are used to denote the behavior elements and Diagonal corners are used to denote the motivation elements [31].

ArchiMate as an Enterprise Architecture modeling language supports a concept of views and viewpoints as defined in the Architecture Description standard [33]. A view is defined as a part of an Architecture Description that addresses a set of related concerns and is tailored for specific stakeholders. The view is specified by means of a viewpoint, which prescribes the concepts, models, analysis techniques, and visualizations that are provided by the view [31].

ArchiMate can be easily extended. There are two mechanisms to extend the language, i.e. Profiles and Specialization. In Profiles, specific attributes of elements and relationships can be created, either in a predefined sense (i.e., as a standard) or in a user-defined sense (i.e., added to cover a specific situation). Any element or relationship in the ArchiMate language can be specialized. A number of suggested specializations are included in the specification. Specializations can associate their own iconography and/or be shown with the UML stereotype convention (<< >>). Using this extension mechanism, it is possible to use the ArchiMate language to model domains that widen the context of business issues within Enterprise Architecture, and that are not focused on IS/IT. An example of such use is given in [7] where ArchiMate is used for modeling Enterprise Risk Management and Security.

2.3 Related Work

The ArchiMate modeling language was developed for modeling Enterprise Architectures on various layers like the Business, Application, and Technology layer, inter-connecting these layers through services, modeling specific viewpoints for selected stakeholders, etc. Moreover, ArchiMate has also been utilized for modeling within other domains. Following the aim of Enterprise Architecture to be a tool for business-IT alignment, ArchiMate was used, for example, for modeling Business Model

Canvas [34,35]. In [36], three distinct enterprise modeling languages, i.e. the Business Model Canvas, E3value, and the business layer of the ArchiMate language were integrated using the ontological schemas.

In the IT Governance area, ArchiMate was used for modeling the ITIL Business Motivation Model [37] and TIPA (Tudor's ITSM Process Assessment) framework [38]. The aim of TIPA [39] is to assess the maturity of specific IT processes based on the ITIL and provide roadmaps for process improvement. The usage of ArchiMate improved the alignment between service management and organization's concepts and further linked ITIL assessments to Enterprise Architecture principles and concepts. In the subsequent paper [40], the COBIT 5 Process Assessment Model (PAM) and TIPA for ITIL v3 were mapped, modeled and integrated using the ArchiMate modeling language with the aim to make the assessment of both frameworks easier.

Further, several papers [41,42] aimed to link business process models mainly modeled in BPMN to Enterprise Architecture models in ArchiMate. Recently, ArchiMate was also used for modeling the Essence, Kernel and Language for Software Engineering Methods [43] and on top of that technological model of disability [44]. However, as to date, ArchiMate has not been used to model international standards.

2.4 Benefits of Using ArchiMate to Model ISO/IEC 29110 Standard

Modeling the standard in ArchiMate is supposed to have both general benefits of visualization, and specific benefits associated with the ISO/IEC 29110 standard. As to general benefits, modeling represents an essential tool for coping with complexity. It enables, among other things, to view a system from multiple perspectives, improve system understanding, discover causes and effects using model traceability, and identify potential consequences of a change. Other benefits come from the use of the modeling tools itself. A number of tools for ArchiMate modeling is available both open source and commercially. An overview of preferable ArchiMate modeling tools can be found in[45], overview of open source tools then in [46]. The ArchiMate modeling language has become supported also by software tools intended for UML and BPMN modeling, e.g. Visual Paradigm [47], Modelio [48] and Signavio [49]. A use of such tools enables modeling or viewing not only ArchiMate models, but also UML or BPMN models, in case they are used in a company, by using the same tool. ArchiMate, in fact, has not been developed to replace UML and BPMN languages, but to complement them. The strength of ArchiMate lies in inter-relating various layers like Business, Application, and Technology through services, modeling various states like as-is and to-be, modeling motivation and strategy. UML is still used for modeling software product development and BPMN for business process modeling. The recommended modeling tool for ArchiMate modeling as such, is Archi [50], which is the world's most popular open source ArchiMate modeling tool. Thus, also the models presented in this paper were developed in Archi. Using modeling tools enhances the benefits of visual

modeling. Once the modeling tool and modeling notation are mastered, the process of making and changing diagrams is quick and agile [32]. Diagrams provide an efficient way to communicate ideas among team members and different stakeholders. As presented in [51], ArchiMate contributes to an effective collaboration over the different domains and has the potential to support shared understanding among stakeholders by creating the required views that represent the viewpoints of the stakeholders. ArchiMate tools mostly use repository for recording model elements. This enables model analysis, re-use and further documentation. The Open Group has published the ArchiMate Model Exchange File Format standard [5] which supports interoperability between ArchiMate tools and can be used also for format translation between modeling notations, e.g. from ArchiMate notation into UML. Many modeling tools are available on-line as cloud-based software, which enables to work together with colleagues from anywhere.

As to specific benefits of using ArchiMate for modeling specifically the ISO/IEC 29110 standard, they are of two kinds, i.e. the benefits for standard developers and benefits for standard users. The main benefit for standard developers lies in cataloging of elements and ensuring a consistency among all parts of the standard and supporting materials like Deployment Packages. The fact that all diagrams presented in all parts of the standard and Deployment Packages are depicted in the standard notation is also very important. Further, the concept of views enables to model the standard in various levels of detail and from various viewpoints. The use of modeling tools then facilitates tracing of elements through layers and views and helps in ensuring the consistency. In addition, using the motivation elements enables to visualize objectives and goals and ensure their realization. Finally, the ArchiMate's support for modeling multiple layers represents a possibility to model software engineering, systems engineering and service areas of the ISO/IEC 29110 standard in a unified form. ArchiMate evinces a high potential especially for modeling services within the ISO/IEC 29110 standard as they are a key concept of ArchiMate.

The principal benefits for standards users, i.e. Very Small Entities and assessors, lie in capturing standard structure in a unified form, using the same notation and same modeling tool for all the diagrams, and creating views for various stakeholders.

3. Methodology

To fulfill the purpose of the research, a specific methodology was defined, that is at a high level presented in Table 2. Structure of the table was derived from [52]. Four sequential activities are defined and for each activity, a used method, output and evaluation method are specified.

Table 2 Research Method

Order	Activity	Method	Output	Evaluation

				Method
1	Problem Identification	Literature Review Personal Experience	Problem Definition	Expert Interview
2	Basic Profile Metamodel Development	Conceptual Modeling	Basic Profile Metamodel	Model Review
3	Mapping Definition of Basic Profile Metamodel Elements to ArchiMate Language	Ontological Mapping	Basic Profile Metamodel to ArchiMate Mapping	Using BWW Model Prototypical Instantiation
4	Basic Profile ArchiMate Model Development	ArchiMate Modeling Prototypical Instantiation of Proposed Mapping	Basic Profile ArchiMate Model	Expert Interviews

First, the problem was identified based on the literature review focused on the ISO/IEC 29110 standard implementation, personal experience with the ISO/IEC 29110 standard education and localization and hands-on personal experience gained from the supervision of the conducted pilot project of the Basic Profile Project Management Deployment Package implementation in 2017. The arising problems were confirmed through an expert interview with the past pilot project implementer, when he identified ineffectiveness in the implementation process and several inconsistencies between the deployment package and the standard.

Second, the ISO/IEC 29110 Basic Profile Metamodel was developed using conceptual modeling. Third, the mapping of the ISO/IEC 29110 Basic Profile to the ArchiMate language was defined using ontological mapping and evaluated through the Bunge-Wand-Weber (BWW) model. Fourth, the ISO/IEC 29110 Basic Profile ArchiMate Model was developed as a prototypical instantiation of the proposed mapping. The model was then evaluated through expert interviews. The individual research activities are in detail described in the following sections.

4. Basic Profile Metamodel Development

Although ArchiMate is precisely described by its metamodel, the ISO/IEC 29110 standard does not include any metamodel in its contents. Kabaale et al. [53] present an axiom-based metamodel for software processes formalization and illustrate such by modelling the ISO/IEC 29110 processes. However, being abstract for the purpose of mapping to ArchiMate, a concrete metamodel for the ISO/IEC 29110 standard was required to be developed. In view of the fact that the ISO/IEC 29110 standard has a lot of parts as stated in Section 2.1, mapping was thus limited to the mostly used Basic

Profile for software engineering to demonstrate proof of concept. Therefore, mainly Part 2-1: Framework and Taxonomy [11], Part 4-1: Software Engineering - Profile Specifications: Generic Profile Group [16] and Part 5-1-2: Management and Engineering Guide: Generic Profile Group: Basic Profile [17] were reviewed and elements of the Basic Profile were identified, consolidated and described. These elements were also consolidated with the Basic Profile Deployment Packages. The developed metamodel for the ISO/IEC 29110 Basic Profile is presented in Figure 6. The metamodel is developed in the UML class diagram notation following the mostly used notation for meta-modeling and the fact that ArchiMate metamodels are also presented in this notation. The metamodel was then carefully reviewed and verified with the content of the Basic Profile and its Deployment Packages.

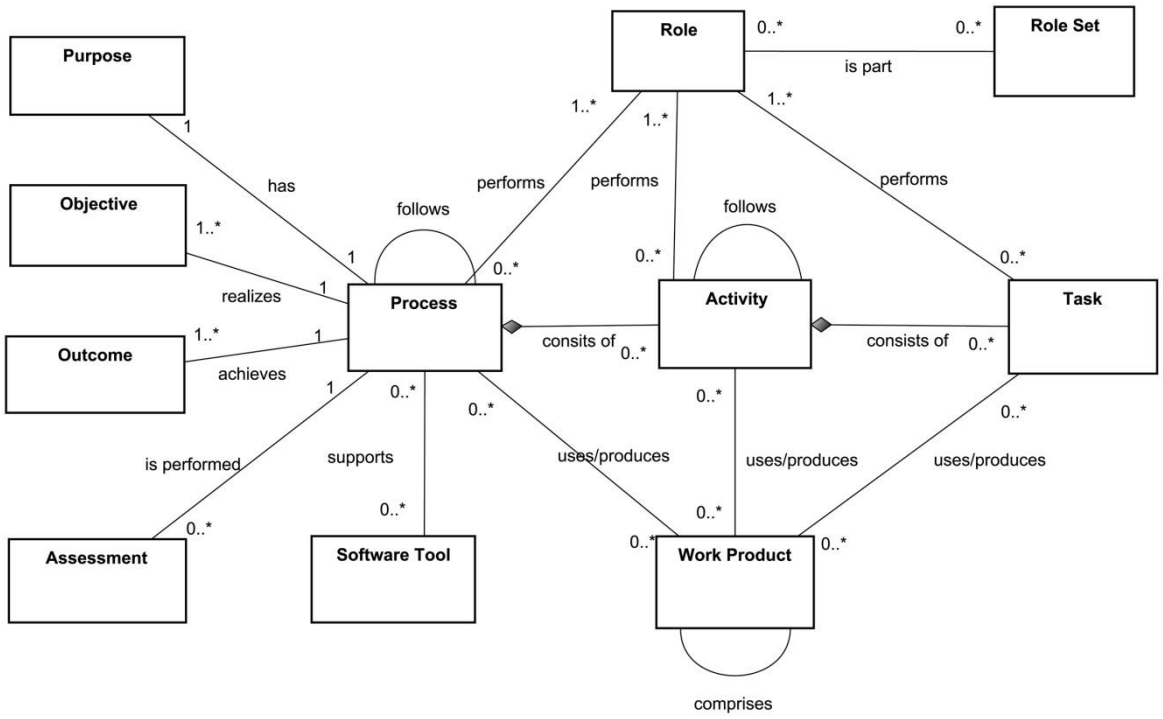


Fig. 6. ISO/IEC 29110 Basic Profile Metamodel

5. Mapping Basic Profile Elements to ArchiMate

For the process of mapping, the latest version 3.0.1 of the ArchiMate language was used [31]. First, available options for modeling the Basic Profile elements in the ArchiMate language were considered being the following [7]:

1. The use of ArchiMate elements unmodified as specified in the standard.

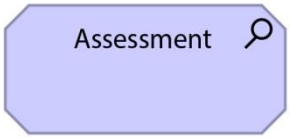
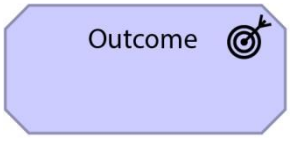
2. The use of the language customization mechanisms as specified in the standard to define additional attributes or specializations of existing ArchiMate elements.

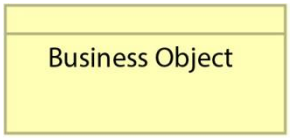
3. The use of additional elements that do not yet exist in the ArchiMate Specification and can be directly linked to existing elements.

The analysis of the developed Basic Profile metamodel showed that unmodified ArchiMate elements are sufficient for representing the Basic Profile elements and relationships. These ArchiMate elements are presented in Table 3 and relationships in Table 4. In one case, a stereotype representing a language customization mechanism was used. It is the case of the Purpose Basic Profile element which is mapped to the Goal ArchiMate element with the stereotype << Purpose >>. The intention behind this customization was to clearly distinguish between the Goal and Purpose Basic Profile elements.

All classes of the Basic Profile metamodel shown in Figure 6 are stated in Table 3 in an alphabetic order and completed with their description derived from the ISO/IEC standard and proposed ArchiMate element (Name and Notation). Mostly, the ArchiMate Business Layer elements (in yellow) are used in the mapping supplemented by the Motivation elements (in violet) and one Application Layer element, i.e. Application Component (in blue). For the Purpose Basic Profile element the Goal ArchiMate element with the stereotype << Purpose >> is used.

Table 3 Mapping Basic Profile Elements to ArchiMate

Basic Profile Element	**Description**	**ArchiMate Element Name**	**ArchiMate Element Notation**
Activity	a set of cohesive tasks of a process	Business Process	
Assessment	disciplined evaluation of an organizational unit's processes against a process assessment model	Assessment	
Objective	specific goal to ensure the accomplishment of the process purpose	Goal	
Outcome	observable result of the successful achievement of the process purpose	Outcome	
Process	a set of interrelated or interacting activities that use inputs to deliver an intended result	Business Process	
Purpose	general goal and result expected of the effective implementation of the process	Goal << Purpose >>	

Role	name and abbreviation of the function to be performed by project team members	Business Role	
Role Set	a set of roles	Business Collaboration	
Software Tool	software application used for providing a process	Application Component	
Task	requirement, recommendation, or permissible action, intended to contribute to the achievement of one or more outcomes of a process	Business Function	
Work Product	artefact associated with the execution of a process	Business Object	

In the ISO/IEC 29110 standard, groups of Work Products are used, i.e. Input Products, Internal Products and Output Products, that are not depicted in the Basic Profile metamodel. These groups are modeled by the ArchiMate element Group (see Figure 7) which denotes a collection of objects that belong together but are not aggregated or composed as with the Grouping element [31].

Fig. 7. ArchiMate Group Element

On the other hand, the Grouping ArchiMate element is used for a collection of the Deployment Packages intended for a specific profile. In Figure 8 Deployment Packages for the Software Engineering Basic Profile are depicted.

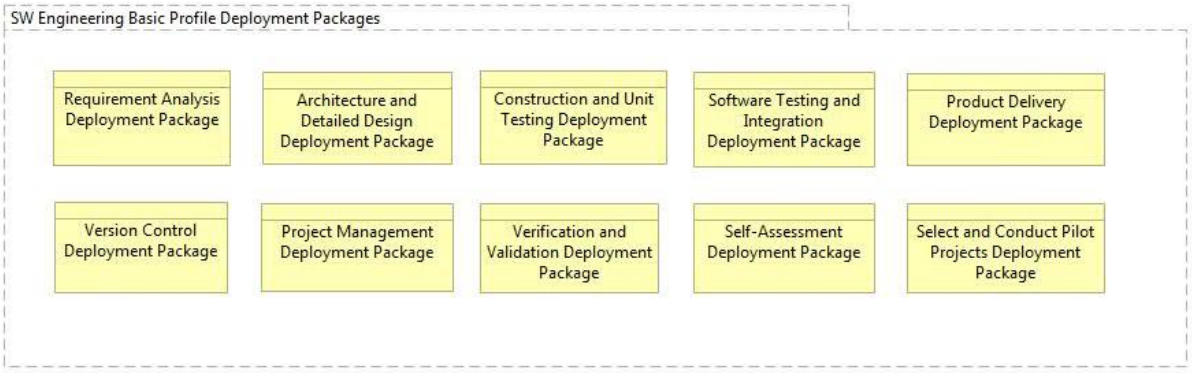


Fig. 8. Deployment Packages modeled as ArchiMate Grouping Element

In Table 4, all relationships of the Basic Profile metamodel described by the Basic Profile relation name, Basic Profile source and target elements are mapped to the ArchiMate relations. These are the strongest relations as ArchiMate enables a derivation of relationships.

Table 4 Relationships Mapping

Relation Name in Basic Profile Metamodel	**From Basic Profile Element**	**To Basic Profile Element**	**Strongest ArchiMate Relation**	**ArchiMate Relation Notation**
has	Process	Purpose	Realization	
has	Process	Objective	Realization	
achieves	Process	Outcome	Realization	
is performed	Assessment	Process	Association	
consists of	Process	Activity	Composition	
supports	Software Tool	Process	Serving	
consists of	Activity	Task	Composition	
performs	Role	Task	Assignment	
is part	Role	Role Set	Aggregation	
uses/produces	Task	Work Product	Access	
comprises	Work Product	Work Product	Composition	
performs	Role	Process	Assignment	
performs	Role	Activity	Assignment	
uses/produces	Process	Work Product	Access	

uses/produces	Activity	Work Product	Access	
follows	Process	Process	Triggering	
follows	Activity	Activity	Triggering	

The ISO/IEC 29110 Basic Profile Metamodel, which was presented in UML Class diagram notation in Figure 6, is then depicted in ArchiMate notation using the proposed mapping as presented in Figure 9.

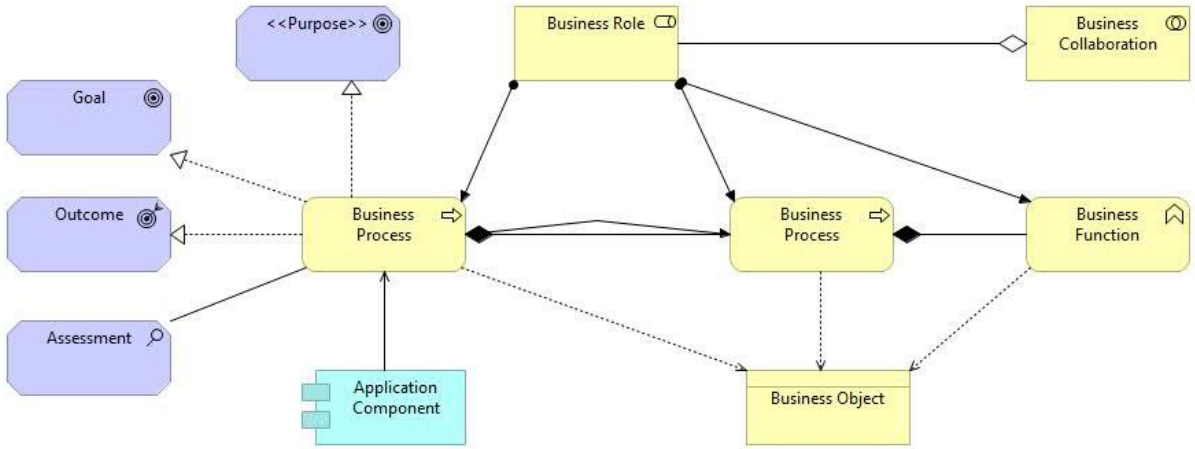


Fig. 9. ISO/IEC 29110 Basic Profile Metamodel modeled in ArchiMate

6. Mapping Evaluation through BWW Model

To evaluate the ISO/IEC 29110 Basic Profile mapping to ArchiMate, the ontological evaluation method of the Bunge-Wand-Weber (BWW) model [54] was selected and utilized. Wand and Weber [54] have extended the systems ontology presented by Mario Bunge [55] and developed a formal foundation called the Bunge-Wand-Weber (BWW) model for modeling information systems [54,56,57]. The BWW model is a high-level ontology containing general concepts that are necessary for description of information systems [58].The description of the BWW elements can be found in [59]. The BWW model, particularly a representation model, has been used in a number of studies for the evaluation of modeling techniques as described in [60].

Mapping the ISO/IEC 29110 Basic Profile to ArchiMate was analyzed based on Ontological Completeness and Ontological Clarity defined by Weber [57] as shown in Figure 10.

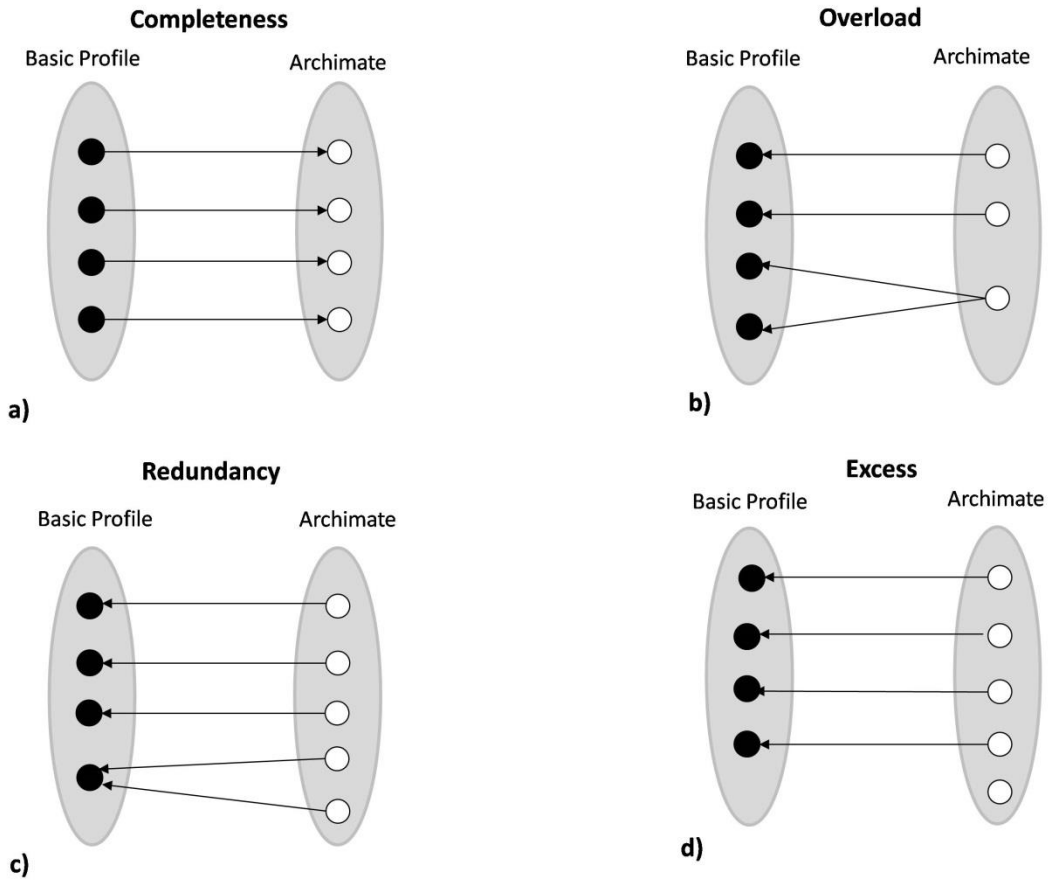


Fig. 10. Ontological Evaluation based on [61]

Completeness is achieved when the mapping between the ISO/IEC 29110 Basic Profile and ArchiMate is total (Figure 10.a) which means that each element of the Basic Profile can be mapped to the ArchiMate element. Clarity is the combination of Overload, Redundancy and Excess of elements. Overload exists when one ArchiMate element is used to model two or more Basic Profile elements (Figure 10.b). Redundancy, on the contrary, occurs when two or more ArchiMate elements are mapped to the same Basic Profile element (Figure 10.c). Excess means that ArchiMate has elements which do not map to any Basic Profile element (Figure 10.d). Overall, incompleteness would be a serious issue for the mapping together with the lack of clarity making reverse mapping difficult.

The results of the mapping evaluation between the ISO/IEC 29110 Basic Profile and ArchiMate are described in this paragraph. As all Basic Profile elements can be mapped to ArchiMate 3.0.1, the proposed mapping is complete. The proposed mapping does not contain any redundancies. As for overload, in a few cases, different Basic Profile elements map onto the same ArchiMate element. This situation occurs in the case of the Process and Activity Basic Profile elements which both map to Business Process. This overload could be reduced using the stereotype. To express the Basic Profile Activity element the ArchiMate Business Process element with << Activity >> stereotype could be

used. Taking into account that the Activity element in the ISO/IEC 29110 Basic Profile is intended to work rather as a sub-process, a clear Business Process ArchiMate element is proposed to be used.

The second case of the overload relates to the Process Purpose and Process Objective Basic Profile elements which both map to Goal. In this case, stereotype << Purpose >> for the ArchiMate Goal element is proposed to represent the Process Purpose Basic Profile element whereas the ArchiMate Goal element stays for the Process Objective Basic Profile element. The intention is to clearly distinguish between the Objective and Purpose Basic Profile elements.

As ArchiMate 3.0.1 is a rich modeling language, it has, of course, a number of excess elements. However, these excess elements do not cause any problems in the case of transformation from the Basic Profile into the ArchiMate model. On the contrary, excess elements of ArchiMate do support modeling other aspects and domains of the ISO/IEC 29110 standard in the ArchiMate language, e.g. a use of the Motivation, Strategy, Implementation and Migration elements.

7. Basic Profile ArchiMate Model Development

This section describes the development of the ISO/IEC 29110 Basic Profile ArchiMate Model that represents a prototypical instantiation of the proposed mapping of the Basic Profile Metamodel to ArchiMate. The Basic Profile for software engineering is in detail described in the Management and Engineering Guide [17]. Besides the graphical representation of the ISO/IEC 29110 series structure encompassed in the standard, the Management and Engineering Guide contains only 3 figures. First one is presented in Figure 11 and shows the Basic Profile processes. The second figure of the original standard is presented in Figure 13 and shows a more detailed decomposition of the Project Management Process. The last figure of the original standard shows then a similar decomposition of the Software Implementation Process, which is not included in this paper. These three figures try to visualize the content of the whole Basic Profile while no standard notation is used for these figures.

The high level view of the Basic Profile developed using ArchiMate is depicted in Figure 12, corresponding to Figure 11 contained in the standard. The Basic Profile has two processes, i.e. Project Management and Software Implementation. The key Work Products Statement of Work is presented as an input and Software Configuration as an output.

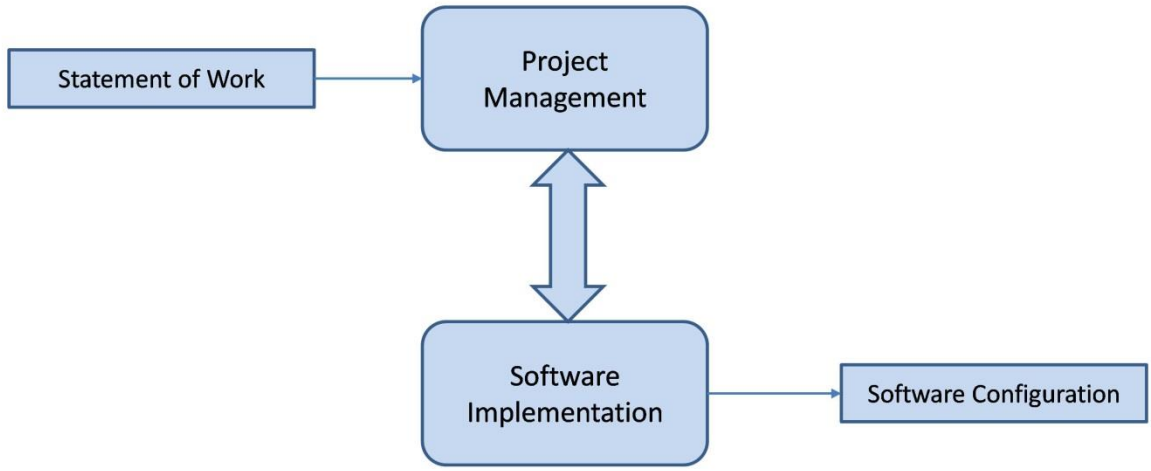


Fig. 11. Basic Profile Processes based on [17]

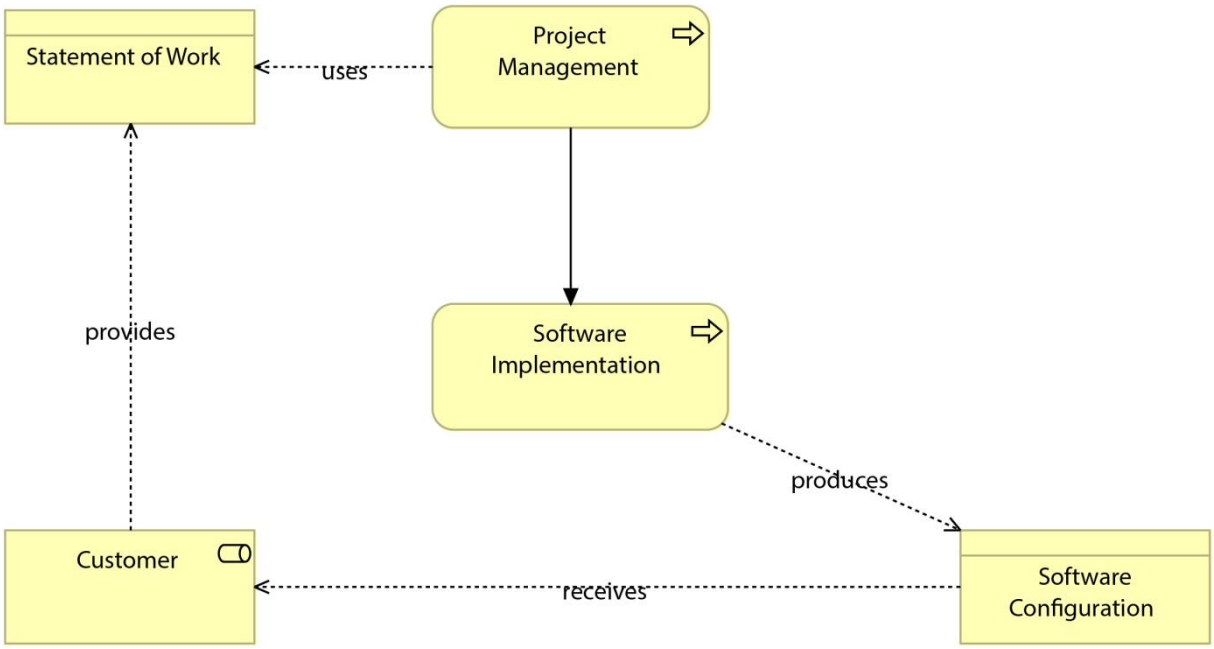


Fig. 12. Basic Profile, High Level View

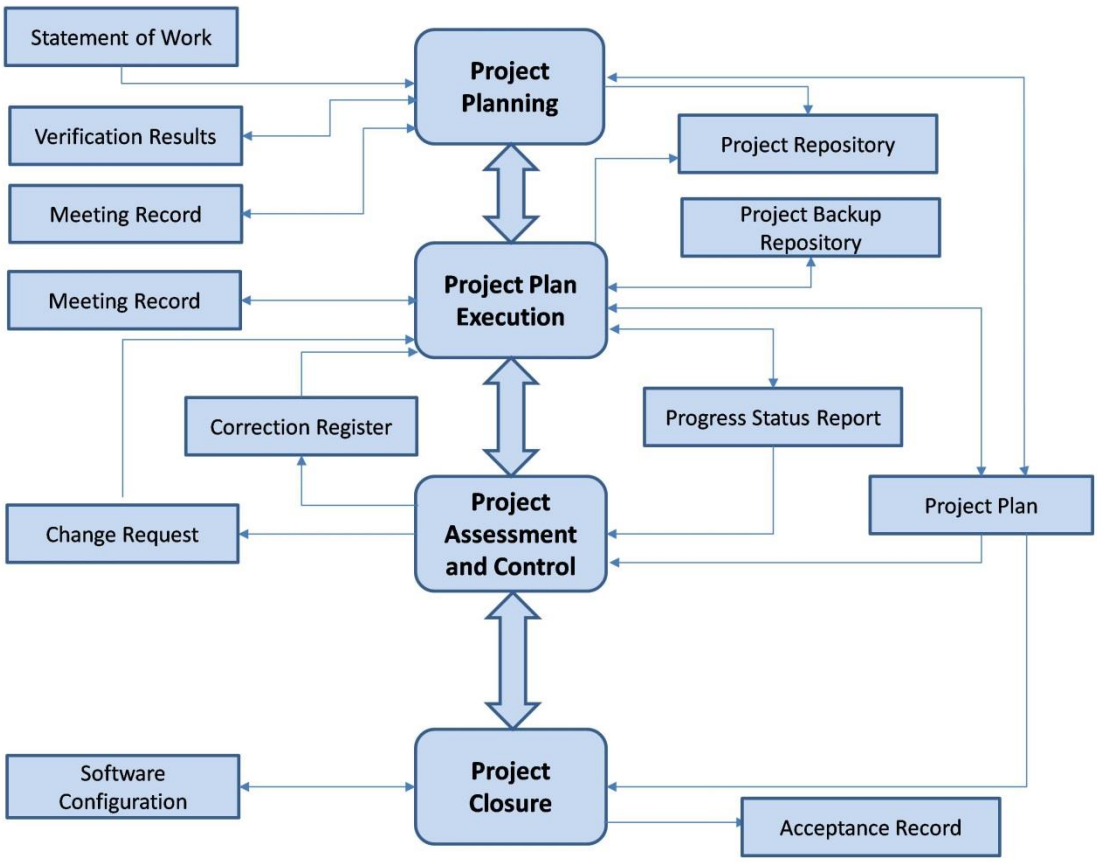


Fig. 13. Project Management Process of the standard based on [17]

In Figure 14, the Project Management Process is depicted, corresponding to Figure 13 contained in the standard. The Project Management Process has in composition four activities: PM 1 Project Planning, PM 2 Project Plan Execution, PM 3 Project Assessment and Control, PM 4 Project Closure. All Work Products used by the Project Management Process are pictured grouped into the Input, Internal, Input and Output and Output Products groups. The Access relation is used between the Process and Work Products. The arrow direction expresses if the Process accesses the Work Product for read, write or read/write.

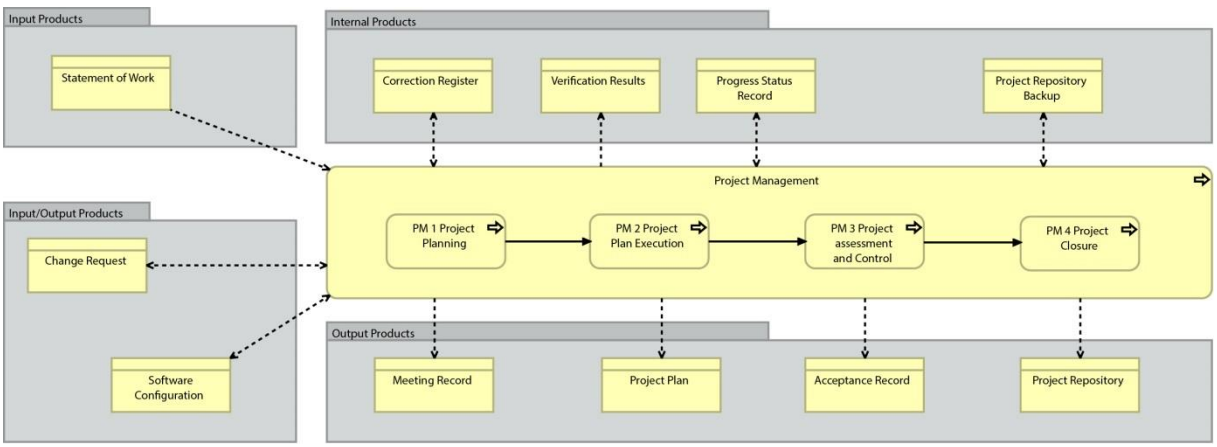


Fig. 14. Project Management Process View

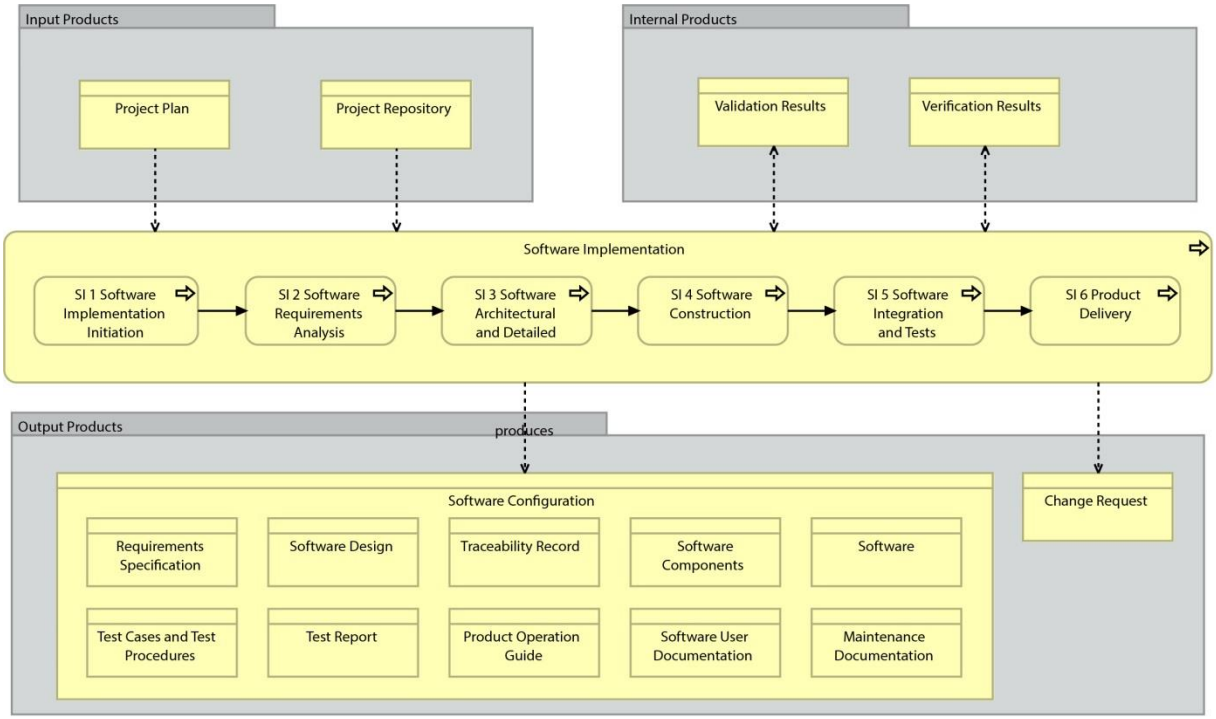


Fig. 15. Software Implementation Process View

Figure 15 then depicts Software Implementation Process of the Basic Profile comprising six activities: SI 1 Software Implementation Initiation, SI 2 Software Requirements Analysis, SI 3 Software Architectural and Detailed Design, SI 4 Software Construction, SI 5 Software Integration and Tests, SI 6 Product Delivery. All work products used by the Software Implementation Process are grouped into the Input, Internal and Output products groups.

Figure 16 shows the objectives of the Project Management Process whereas Figure 17 shows the objectives of the Software Implementation Process. The Process Objectives are represented as the Goal ArchiMate elements. These views show illustratively, which Activities contribute to fulfilling particular objectives. The ArchiMate Realizes relation is used to express this relationship.

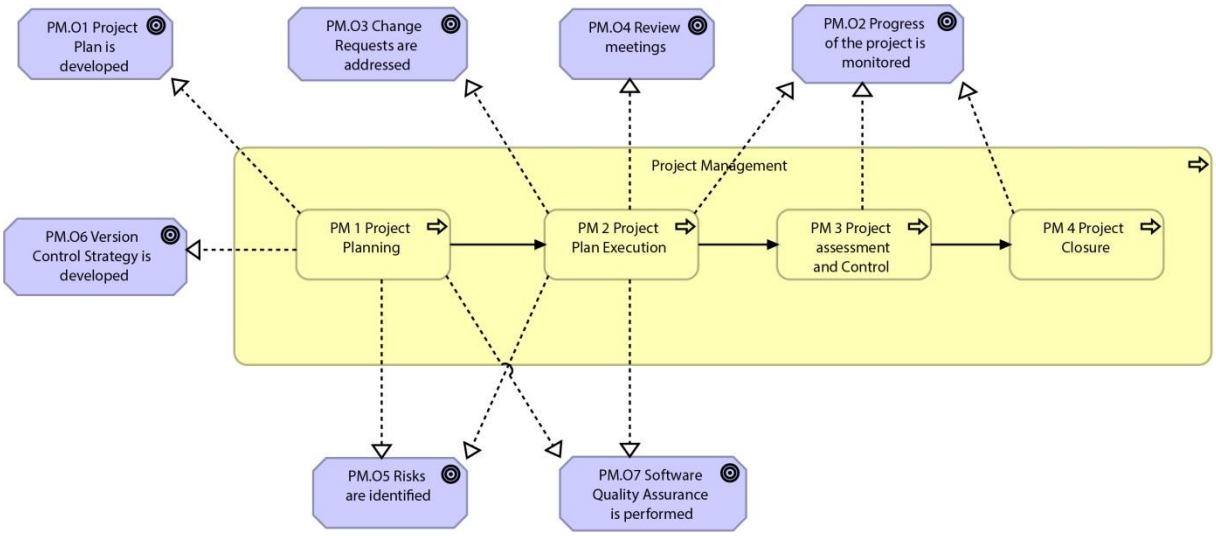


Fig. 16. Objectives of Project Management Process View

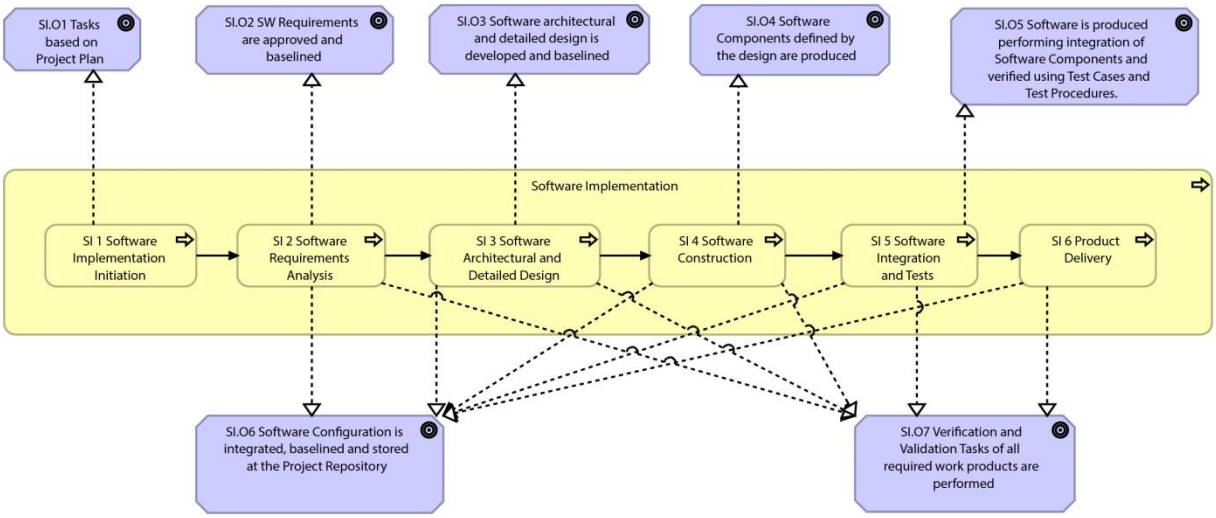


Fig. 17. Objectives of Software Implementation Process View

Figure 18 depicts the Project Planning Activity and its Tasks. Tasks are modeled in ArchiMate as Business Functions. Figure 19 shows a part of a view (cut-out) on an activity level. The Assignment relation is used to depict Roles performing Tasks. In this Figure, the Assignment relation is used between the Business Collaboration element and the Task which expresses that more than one Role participates on performing the Task. In this view, the Work Products used by Tasks are depicted as well. The Access relation is used again to show which Work Products are used in Tasks for read, write or read/write.

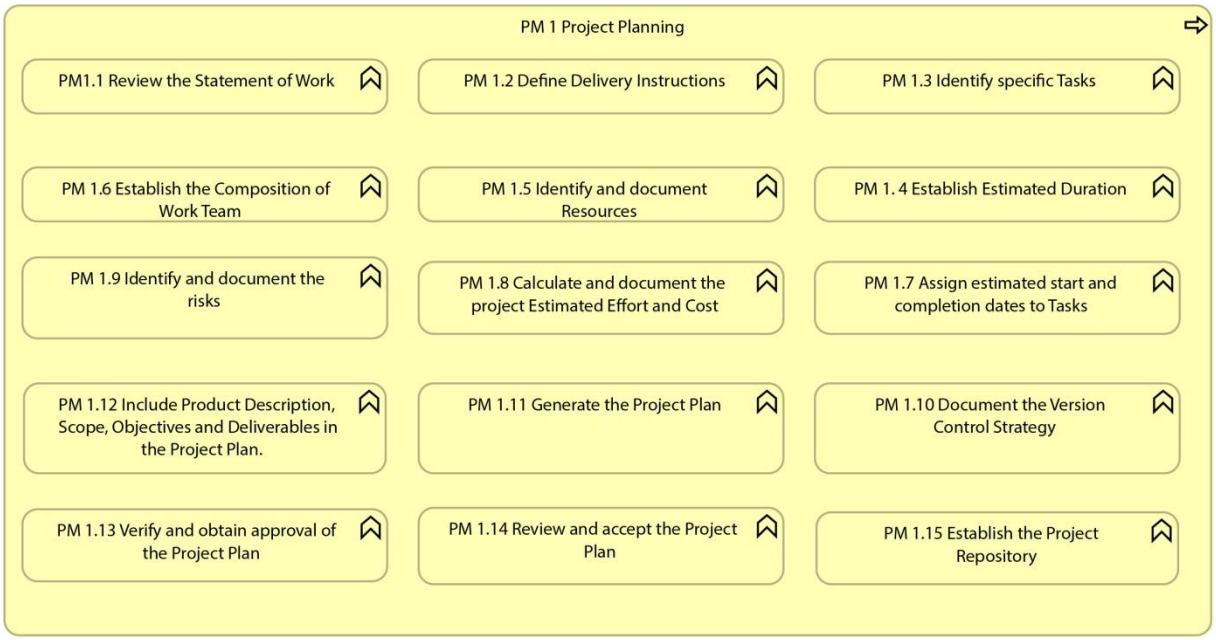


Fig. 18. Tasks of Project Planning Activity

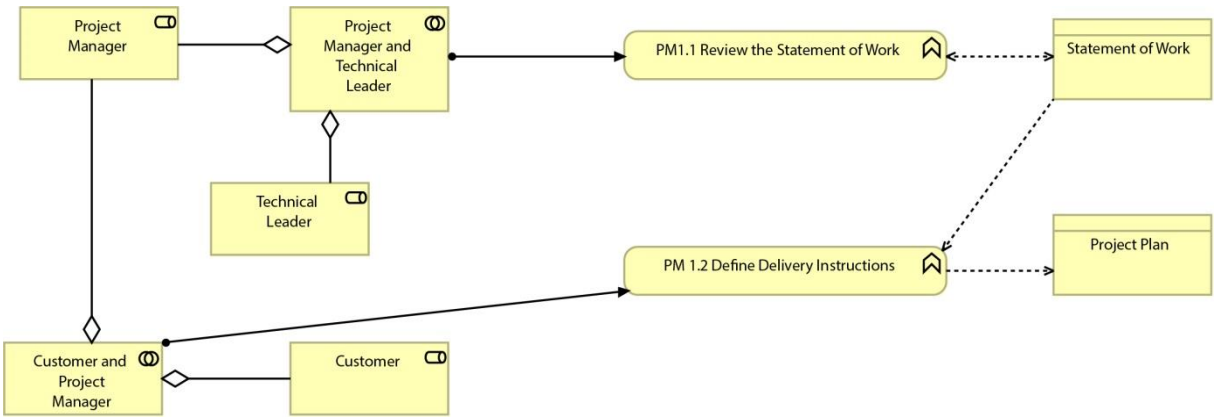


Fig. 19. Part of Project Planning Activity View

As mentioned above, the Deployment Packages were developed to support the implementation of the Profiles in Very Small Entities. Ten Deployment Packages were developed for the Basic Profile which can be modeled in ArchiMate as well. This way it is possible to reuse the ArchiMate Basic Profile elements in Deployment Packages and ensure consistency. In Figure 20, the Select and Conduct Pilot Projects Deployment Package is modeled, which was developed to support conducting pilot projects on a particular profile or a deployment package implementation as mentioned in Section 2.1. This Deployment Package has one Process consisting of four Activities and their Tasks, Roles assigned to this Process and Work Products used.

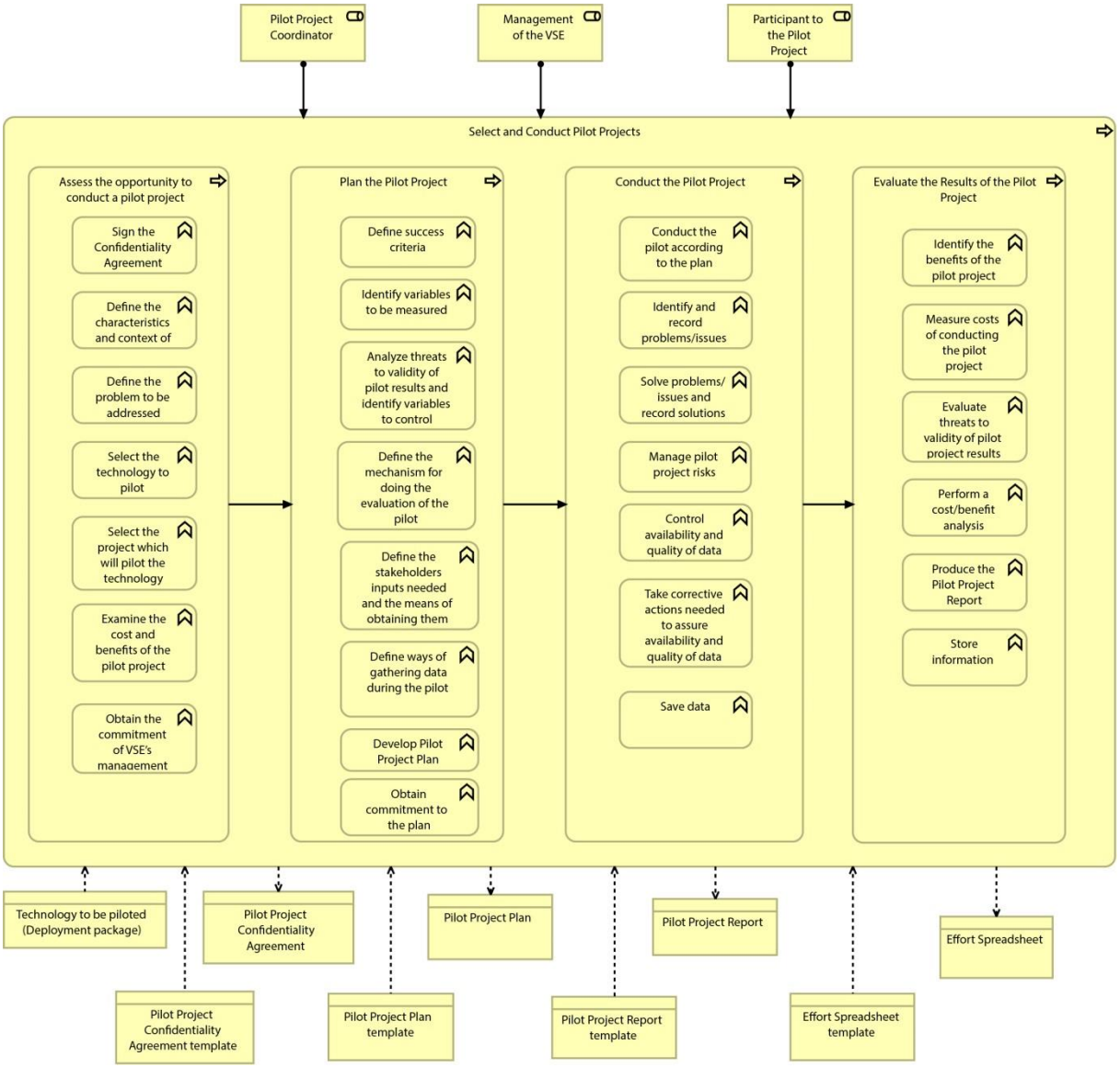


Fig. 20. Conduct Pilot Projects Deployment Package View

8. Basic Profile ArchiMate Model Evaluation

The ISO/IEC 29110 Basic Profile ArchiMate Model developed as the prototype instantiation of the proposed mapping was evaluated through expert interviews. Expert interviews were selected as a suitable evaluation method as this method proves whether an artefact solves a specific problem [52]. This method was used for example for the evaluation of a social program [62]. First, the preparation of the expert interviews is described and then the results of the interviews are analyzed.

8.1. Expert Interviews Preparation

To conduct the model evaluation, the evaluation question was defined as follows: Evaluate that the ISO/IEC 29110 Basic Profile ArchiMate Model positively supports and facilitates the Basic Profile implementation, show potential challenges of this model and opportunities for an improvement.

Then, the experts were selected based on purposive sampling [63]. In Table 5, the sample of selected experts is presented with the indication why they were selected and if they have knowledge of the ISO/IEC 29110 Basic Profile and ArchiMate notation.

Table 5 Selected Interview Partners

Expert	Grounds for Selection	Knowledge of ISO/IEC 29110 Basic Profile	Knowledge of ArchiMate
#1	Is performing software development in a Very Small Entity	Yes	No
#2	Was conducting a pilot project on the implementation of ISO/IEC 29110 Basic Profile in 2016	Yes	No
#3	Will conduct a pilot project on the implementation of ISO/IEC 29110 Basic Profile in 2019	Yes	Yes
#4	Was implementing the ISO/IEC 29110 in Eclipse Process Framework Composer	Yes	No

To acquire expert opinions, a qualitative semi-structured interviewing was applied. First, the intended topic areas of the interviews were established based on the evaluation question, i.e. (1) the challenges in the ISO/IEC 29110 Basic Profile implementation; (2) how the Basic Profile ArchiMate Model facilitates the Basic Profile implementation; (3) the perception of the Basic Profile ArchiMate Model problem areas; (4) the opportunities for an improvement in the experts' view. The most important topic area in the evaluation process represents the second area which evaluates the benefits of the Basic Profile ArchiMate Model to the implementation of the Basic Profile or pilot projects. To identify evaluation questions within this area, potential benefits of using ArchiMate for the Basic Profile modeling were taken into account. These potential benefits were described in Section 2.4. From the perspective of the expert interviews evaluation, general benefits of modeling and specific benefits for standards users, i.e. Very Small Entities and pilot project coordinators, were considered. In total, 15 questions were prepared covering all four topic areas.

At the beginning of the interview, the evaluation design was briefly outlined and subject of the evaluation presented, i.e. the current form of the ISO/IEC 29110 Basic Profile, ArchiMate language, the proposed mapping of the Basic Profile to ArchiMate and the ISO/IEC 29110 Basic Profile ArchiMate Model implemented in the ArchiMate modeling tool. The interviews lasted 1 hour in duration and were audio recorded. To interview one expert (#3), a written form of the questionnaire was used as the expert is at a long-term stay overseas and thus could not have been reached personally.

8.2. Expert Interviews Results

The results from the expert interviews are presented in this section and further analyzed, Overall, the interviewees agreed that the text of the standard itself is quite brief and accurately written, but it is rather difficult to understand from the text itself what needs to be implemented and in which way and which parts are interconnected. From the responses to the evaluation questions within the key topic area of how the Basic Profile ArchiMate Model facilitates the Basic Profile implementation, several key themes were extracted and the experts' evaluation captured in Table 6.

Table 6 Basic Profile ArchiMate Model Benefits for Standard Implementation

Benefits	Interviewee #1	Interviewee #2	Interviewee #3	Interviewee #4
Advantages of larger visualization	definitely	definitely	definitely	definitely
Helps understanding of context and relationships	definitely	definitely	definitely	definitely
Easier customization of the Basic Profile	definitely	probably	probably	definitely
Improving communication within the team and with other stakeholders	definitely	definitely but model itself is not enough	definitely but employees need to know syntax	definitely

Following the evaluation questions and evaluating potential benefits of the Basic Profile ArchiMate Model, the interviewees responded that they were not able to realize the complexity and interconnectivity of the standard after only reading the standard as it is. Although they acquired information on what is supposed to be done when implementing the standard, they were not able to perform concrete steps and follow the processes. The content of the Basic Profile modeled in ArchiMate helps to understand the context and relationships between various parts of the standard. The visualization that the model brings shows concrete steps for the implementer to follow and

supports the implementation across a VSE by enabling all stakeholders to see, understand and follow a concrete implementation plan. Based on the responses, it is also appropriate that the content used by the stakeholders should be interactive and they should be able to click through it being more comprehensible.

Regarding customization of the Basic Profile elements to the needs of VSEs, the interviewees assessed this possibility as very beneficial. Within a VSE, each person has a different role and responsibilities and nobody wants to or is able to (with regards to limited time) do an extra work. Thus, if the employees receive a filtered customized content in the form of a view, they are more likely to use it. The more the processes match the exact way a VSE works the better.

Regarding localization of the standard and deployment packages, the interviewees proclaimed that for an employment of the standard in the Czech Republic it is of high importance to have the standard localized in the Czech language. The Basic Profile ArchiMate Model is able to help with such localization.

Evaluating the impact of the Basic Profile ArchiMate Model on a communication within the team and with other stakeholders, the interviewees agreed that the graphical form is always more comprehensible than just a text, and thus it simplifies and streamlines communication. However, the employees need to know the basic syntax (terminology) of ArchiMate models or have someone on hand to explain them the model.

Assessing the fact that the Basic Profile is modeled in a standard notation, the interviewees agreed that the notation is understandable, especially with a legend on hand. However, it needs to be explained in advance to those stakeholders that are not familiar with the notation. The advantage of using a standard notation lies also in the fact that people are familiar with the notation from school, work or training. At the same time, its knowledge enhances employability in the labor market.

The possibility that a VSE will use ArchiMate also for modeling the software product being built, e.g. its application architecture, products, business processes at a customer, is not considered as beneficial from the view of the VSE representative as they do not model software products in their company, but write the code directly. On the other hand, other interviewees consider the possibility that employees will learn to model and then use the models also for the software product development to be beneficial.

Among advantages of using the Basic Profile ArchiMate Model, the interviewees stated effortless archiving, reusability, sharing models within an enterprise and easy and quick changes. Among disadvantages of using the Basic Profile ArchiMate Model, they named the necessity to learn how to use the ArchiMate modeling tool and ArchiMate language and time demands for developing and modifying models.

Evaluating possible larger effectiveness of the standard or pilot project implementation when using the Basic Profile ArchiMate Model, the implementer of the previous pilot project stated that the standard in a text form hinders customization, and thus all aspects that enable customization are beneficial and effective. Improving the understanding of the standard should thus increase the effectiveness of its implementation.

Regarding the problems or risks associated with the usage of the Basic Profile ArchiMate Model, all interviewees agreed that modeling is laborious. However, if the models are prepared and can be modified, the model is efficiently utilizable in daily workload. One interviewee sees the risk in the formality of modeling. As the Basic Profile ISO/IEC 29110 is designed for Very Small Entities of up to 25 people, modeling could become a time consuming process not worth the cost for such a small entity.

Lastly, the interviewees stated suggestions for an improvement of the model such as the need to depict the legend within the models, printing and viewing models in color and having the possibility of a responsive mobile view of the models.

9. Discussion

9.1 Evaluation of ISO/IEC 29110 Basic Profile ArchiMate Model Benefits

The motivation for this research was to facilitate the ISO/IEC 29110 Basic Profile implementation performed by Very Small Entities or pilot projects coordinators using the ArchiMate model of the ISO/IEC 29110 Basic Profile.

The potential benefits of using ArchiMate to model the ISO/IEC 29110 standard were described in Section 2.4. These benefits were confirmed during the ISO/IEC 29110 Basic Profile ArchiMate Model evaluation through expert interviews described in previous section.

The visualization that the ISO/IEC 29110 Basic Profile ArchiMate Model brings shows concrete steps for the project implementer to follow and supports the implementation across a VSE. As the Management and Engineering Guide for the Basic Profile [17] has currently only three pictures capturing the content of the profile. Having the well-known saying "A picture is worth a thousand words" in mind, it is valuable to supplement the standard with more diagrams and offer a visual aspect of the implementation.

To offer VSEs the benefits of customization and effectiveness, the proposed scenario is that the Basic Profile ArchiMate Model will be published on the ISO/IEC 29110 website and freely available. This way VSEs can utilize this model as is, broaden it or customize it to their needs. This would save a great deal of effort during the profile implementation and ensure a proper usage of the standard

elements. To develop implementation plans, the ArchiMate Implementation and Migration elements can also be applied.

Moreover, the ArchiMate modeling language can be utilized to model the system or software product under development as well using the same notation and modeling tools. Using the same modeling notation for all VSE Profiles and Deployment Packages significantly reduces the implementation complexity and increases understanding.

9.2 Limitations and Recommendations for Further Research

To establish the research presented in this paper, a total of two limitations were taken into consideration. First, the mapping was limited to the ISO/IEC 29110 Basic Profile for software engineering due to the limited scope. Selecting just the Basic Profile was based on multiple reasons being as follows. The Basic Profile represents the basis for other Profiles and comprises key Processes, Activities, Roles and Work Products. It is the oldest ISO/IEC 29110 Profile, and the only profile in which a VSE can be certified. Further, the key part of the Basic profile, i.e. Management and Engineering Guide [17] is available free of charge. The highest amount of Deployment Packages has been developed just for the Basic Profile available free of charge as well. This Profile is also very often used within university courses. On top of that, the Basic Profile has been translated into several languages either officially or unofficially.

Second, a comprehensive evaluation of the mapping results, i.e. Basic Profile models in ArchiMate notation has not been performed among Very Small Entities yet. However, conducted expert interviews confirmed the usability of the ISO/IEC 29110 Basic Profile in ArchiMate and its potential to improve efficiency of standard implementation. The dissemination of the proposed mapping both among the ISO/IEC 29110 standard developers, i.e. Working Group 24 within ISO/IEC JTC1 SC7, and standard users, i.e. Very Small Entities and assessors, helps to promote practical evaluation of the Basic Profile models created in ArchiMate among Very Small Entities from various countries worldwide.

The above stated limitations then influence the recommendations for further research. First, an evaluation of the usage of the ArchiMate models in a pilot project planned for 2019 will be performed. Simultaneously, the ISO/IEC 29110 ArchiMate model will be evaluated by the ISO/IEC 29110 standard developers, i.e. Working Group 24 within ISO/IEC JTC1 SC7. To support the evaluation, the author will contribute to making all the ArchiMate models available at the public site of the ISO/IEC 29110 standard [28].

Following the broad evaluation of the ArchiMate models, other ISO/IEC 29110 Profiles should be modeled. To make the utilization of the ArchiMate models of the ISO/IEC 29110 standard easier, a

method describing the usage of the ArchiMate models within implementation and assessment projects should be developed.

Further, modeling of other aspects of the ISO/IEC 29110 standard in the ArchiMate language should be considered, especially other Motivation elements could be used, as well as Strategy and Implementation and Migration elements.

Lastly, the ArchiMate language could be also used for modeling other software and systems engineering standards within other working groups of the ISO/IEC JTC1 SC7. This is in compliance with other activities focusing on a unification of language and terminology used in software development [64], which was identified even within the ISO/IEC JTC1 SC7 [65].

10. Conclusion

The purpose of this paper was to define specific mapping between the ISO/IEC 29110 Basic Profile and ArchiMate, which enabled to develop the ISO/IEC 29110 Basic Profile ArchiMate Model aimed at improving Basic Profile implementation in Very Small Entities. To fulfill the purpose of this paper, several sequential activities being part of the defined methodology were performed.

First, the ISO/IEC 29110 standard elements were consolidated and the metamodel for the ISO/IEC 29110 Basic Profile was developed. After selecting the suitable option for modeling the Basic Profile in ArchiMate, real mapping of the Basic Profile metamodel elements to the ArchiMate language was performed. The proposed mapping was then evaluated using the Bunge-Wand-Weber (BWW) model for completeness and clarity. The mapping was complete in the sense that every Basic Profile element can be mapped to ArchiMate. There were some clarity issues, especially two overload cases that are described and explained, and a number of excess elements which do not cause any problems in the case of transformation from the Basic Profile into the ArchiMate model.

The proposed mapping was then used for modeling the ISO/IEC 29110 Basic Profile and its Deployment Packages. Several views depicting the ISO/IEC 29110 standard from various levels and viewpoints were presented. Finally, the ISO/IEC 29110 Basic Profile ArchiMate Model was developed and evaluated using expert interviews. The results of expert interviews confirmed the beneficial and efficient aspects of the model and its support role in the implementation across VSEs. The benefits of using ArchiMate for modeling the ISO/IEC 29110 standard were stated and research limitations were indicated implying future research.

In conclusion, the contribution of this paper is multiple. First, the ISO/IEC 29110 Basic Profile Metamodel was developed, which can be used by standard developers to ensure consistency of the standard elements and further standard development. Second, as a main output, the mapping between the ISO/IEC 29110 Basic Profile Metamodel and ArchiMate language was defined and evaluated for

completeness and clarity. Third, the proposed mapping was used for modeling the content of the ISO/IEC 29110 Basic Profile in ArchiMate that resulted in the development of the ISO/IEC 29110 Basic Profile ArchiMate Model, which represents a key contribution. The Basic Profile ArchiMate Model was implemented in a freely available modeling tool and can be used by VSEs, assessors and standard developers. The model was preliminarily evaluated using expert interviews. A comprehensive evaluation within a pilot project will be performed in 2019 along with the evaluation by the ISO/IEC 29110 standard developers, i.e. Working Group 24 within ISO/IEC JTC1 SC7.

Acknowledgement

The work reported in this paper was processed with the contribution of the long term institutional support of research activities by the Faculty of Informatics and Statistics, University of Economics, Prague.

References

[1] C.Y. Laporte, M. Munoz, J.M. Miranda, R. V. O'Connor, Applying Software Engineering Standards in Very Small Entities: From Startups to Grownups, IEEE Softw. 35 (2018) 99–103. doi:10.1109/MS.2017.4541041.

[2] C.Y. Laporte, R. V. O'Connor, Implementing Process Improvement in Very Small Enterprises with ISO / IEC 29110 A Multiple Case Study Analysis, (2016). doi:10.1109/QUATIC.2016.27.

[3] L.G. Paucar, C.Y. Laporte, J. Arteaga, M. Bruggmann, Implementation and Certification of ISO/IEC 29110 in an IT Startup in Peru, 17 (2015) 14.

[4] R. V. O'Connor, Early Stage Adoption of ISO/IEC 29110 Software Project Management Practices: A Case Study, in: Int. Conf. Softw. Process Improv. Capab. Determ., 2014: pp. 226–237. doi:10.1007/978-3-319-13036-1_20.

[5] P. Clarke, R. V. O'Connor, B. Leavy, A complexity theory viewpoint on the software development process and situational context, Proc. Int. Work. Softw. Syst. Process - ICSSP '16. (2016) 86–90. doi:10.1145/2904354.2904369.

[6] M. Yilmaz, R. V. O'Connor, P. Clarke, A Systematic Approach to the Comparison of Roles in the Software Development Processes, in: D.A. Mas A., Mesquida A., Rout T., O'Connor R.V. (Ed.), Softw. Process Improv. Capab. Determ. SPICE 2012., Springer, Berlin, Heidelberg, 2012.

[7] I. Band, W. Engelsman, C. Feltus, S.G. Paredes, J. Hietala, H. Jonkers, S. Massart, Modeling Enterprise Risk Management and Security with the ArchiMate Language, Open Gr. (2014) 40.

[8] A. Anacleto, C.G. Von Wangenheim, C.F. Salviano, R. Savi, C. De Pesquisas, R. Archer, Experiences Gained from Applying ISO / IEC 15504 to Small Software Companies in Brazil, in: 4th Int. SPICE Conf. Process Assess. Improv., 2004: pp. 1–5. http://www.inf.ufsc.br/~gresse/download/Experiences_spice04_vref.pdf.

[9] C.Y. Laporte, S. Alexandre, R. V. O'Connor, A Software Engineering Lifecycle Standard for Very Small Enterprises, in: Eur. Conf. Softw. Process Improv., Springer, Berlin, Heidelberg, 2008: pp.

129–141. doi:10.1007/978-3-540-85936-9_12.

[10] ISO/IEC TR 29110-1 Systems and software engineering -- Lifecycle profiles for Very Small Entities (VSEs) -- Part 1: Overview, (2016).

[11] ISO/IEC 29110-2-1 Software engineering -- Lifecycle profiles for Very Small Entities (VSEs) -- Part 2-1: Framework and taxonomy, (2015).

[12] ISO/IEC TR 29110-3-1 Systems and software engineering -- Lifecycle profiles for Very Small Entities (VSEs) -- Part 3-1: Assessment guide, (2015).

[13] ISO/IEC 29110-3-2 Systems and software engineering -- Lifecycle profiles for Very Small Entities (VSEs) -- Part 3-2: Conformity certification scheme, (2018).

[14] ISO/IEC 29110-3-3 Systems and software engineering -- Lifecycle profiles for Very Small Enterprises (VSEs) -- Part 3-3: Certification requirements for conformity assessments of VSE profiles using process assessment and maturity models, (2016).

[15] ISO/IEC TR 29110-3-4 Systems and software engineering -- Lifecycle profiles for Very Small Entities (VSEs) -- Part 3-4: Autonomy-based improvement method, (2015).

[16] ISO/IEC 29110-4-1 Systems and software engineering -- Lifecycle profiles for Very Small Entities (VSEs) -- Part 4-1: Software engineering - Profile specifications: Generic profile group, (2018).

[17] ISO/IEC TR 29110-5-1-2 Software Engineering - Lifecycle Profiles for Very Small Entities (VSEs) - Part 5-1-2: Management and Engineering Guide: Generic Profile Group: Basic Profile, (2011).

[18] ISO/IEC TR 29110-5-1-1 Software Engineering - Lifecycle Profiles for Very Small Entities (VSEs) - Part 5-1-1: Management and Engineering Guide: Generic Profile Group: Entry Profile, (2012).

[19] ISO/IEC TR 29110-5-1-3 Systems and software engineering -- Lifecycle profiles for Very Small Entities (VSEs) -- Part 5-1-3: Software engineering -- Management and engineering guide: Generic profile group -- Intermediate profile, (2017).

[20] ISO/IEC TR 29110-5-1-4 Software and systems engineering-- Lifecycle profiles for very small entities (VSEs) -- Part 5-1-4: Software engineering: Management and engineering guidelines: Generic profile group: Advanced profile, (2018).

[21] ISO/IEC TR 29110-5-2-1 Systems and Software Engineering - Lifecycle Profiles for Very Small Entities (VSEs) - Part 5-2-1: Organizational Management Guidelines, (2016).

[22] C.Y. Laporte, R. V. O'Connor, A Systems Process Lifecycle Standard for Very Small Entities: Development and Pilot Trials, in: 21th Eur. Softw. Process Improv. Conf. (EuroSPI 2014), Springer-Verlag, Heidelberg, 2014: pp. 13–24. doi:10.1007/978-3-662-43896-1.

[23] ISO/IEC TR 29110-5-6-1 Systems and Software Engineering - Lifecycle Profiles for Very Small Entities (VSEs) - Part 5-6-1: Systems Engineering - Management and Engineering Guide: Generic Profile Group: Entry Profile, (2015).

[24] ISO/IEC TR 29110-5-6-2 Systems and Software Engineering - Lifecycle Profiles for Very Small Entities (VSEs) - Part 5-6-2: Systems Engineering - Management and Engineering Guide: Generic Profile Group: Basic Profile, (2014).

[25] S. Galvan-Cruz, M. Mora, R. V. O'Connor, F. Acosta, F. Álvarez, An Objective Compliance Analysis of Project Management Process in Main Agile Methodologies with the ISO/IEC 29110

Entry Profile, Int. J. Inf. Technol. Syst. Approach. 10 (2017) 75–106. doi:10.4018/IJITSA.2017010105.

[26] ISO/IEC TR 29110-5-3 Systems and software engineering -- Lifecycle profiles for Very Small Entities (VSEs) -- Part 5-3: Service delivery guidelines, (2018).

[27] R. V. O'Connor, C.Y. Laporte, The Evolution of the ISO/IEC 29110 Set of Standards and Guides, Int. J. Inf. Technol. Syst. Approach. 10 (2017) 1–21. doi:10.4018/IJITSA.2017010101.

[28] Deployment Packages repository, (2018). https://profs.etsmtl.ca/claporte/English/VSE/index.html (accessed August 17, 2018).

[29] M. Lankhorst, Enterprise architecture at work : modelling, communication and analysis, Springer, 2013.

[30] A. Josey, M. Lankhorst, I. Band, H. Jonkers, D. Quartel, An introduction to the ArchiMate 3.0 Specification, Open Gr. (2016) 1–20.

[31] ArchiMate® 3.0.1 Specification, (n.d.). http://pubs.opengroup.org/architecture/archimate3-doc/ (accessed August 15, 2018).

[32] E. Walters, Understanding the Basics An Introduction to the ArchiMate® Modeling Language, Version 3.0.1, 2017.

[33] ISO/IEC/IEEE 42010 Systems and software engineering -- Architecture description, (2011).

[34] A. Osterwalder, Y. Pigneur, T. Clark, A. Smith, Business model generation : a handbook for visionaries, game changers, and challengers, n.d. https://www.wiley.com/en-us/Business+Model+Generation%3A+A+Handbook+for+Visionaries%2C+Game+Changers%2C+and+Challengers-p-9780470876411 (accessed August 15, 2018).

[35] M.E. Iacob, L.O. Meertens, H. Jonkers, D.A.C. Quartel, L.J.M. Nieuwenhuis, M.J. van Sinderen, From enterprise architecture to business models and back, Softw. Syst. Model. (2012) 1059–1083. doi:10.1007/s10270-012-0304-6.

[36] A. Caetano, G. Antunes, J. Pombinho, M. Bakhshandeh, J. Granjo, J. Borbinha, M.M. da Silva, Representation and analysis of enterprise models with semantic techniques: an application to ArchiMate, e3value and business model canvas, Knowl. Inf. Syst. 50 (2017) 315–346. doi:10.1007/s10115-016-0933-0.

[37] M. Vicente, N. Gama, M. da Silva Mira, Modeling ITIL business motivation model in ArchiMate, in: Int. Conf. Explor. Serv. Sci., 2013: pp. 86–99. doi:10.1007/978-3-642-36356-6.

[38] N. Silva, M. Mira, C.R.P.H. Tudor, Using ArchiMate to Model a Process Assessment Framework, (2015) 1189–1194.

[39] Van Haren Publishing., ITSM Process Assessment Supporting ITIL - Using TIPA to Assess and Improve Your Processes With ISO 15504 and Prepare for ISO 20000 Certification., Van Haren Pub, 2009.

[40] R. Almeida, P.L. Pinto, M. Mira, Using ArchiMate to Assess COBIT 5 and ITIL Implementations, Int. Conf. Inf. Syst. Dev. (2016) 235–246.

[41] L. Penicina, M. Kirikova, Towards Completeness and Lawfulness, Perspect. Bus. Informatics Res. Bir 2013. 158 (2013) 63–77.

[42] L. Penicina, Linking BPMN, ArchiMate, and BWW: Perfect match for complete and lawful business process models?, CEUR Workshop Proc. 1023 (2013) 156–165.

[43] N. Syynimaa, Essence : Reference Architecture for Software Engineering, in: ICEIS 2018, 2018: pp. 345–350. doi:10.5220/0006793603450350.

[44] M.A. Hussain, K. Ahsan, S. Iqbal, A. Nadeem, Disability and Digital Divide : Bridging the Gap Through Archimate Approach, Pak. J. Sci. 68 (2016) 426–432.

[45] Best Enterprise Architecture Software | 2018 Reviews of the Most Popular Systems, (n.d.). https://www.capterra.com/enterprise-architecture-software/ (accessed August 25, 2018).

[46] Open source Enterprise Architecture tools | Inform-IT, (n.d.). https://www.inform-it.org/open-source-enterprise-architecture-tools/ (accessed August 25, 2018).

[47] Visual Paradigm - Leading UML, BPMN, EA, Agile and Project Management Software, (n.d.). https://www.visual-paradigm.com/ (accessed August 31, 2018).

[48] Modelio Open Source - UML and BPMN free modeling tool, (n.d.). https://www.modelio.org/ (accessed August 31, 2018).

[49] Collaborative process & decision management | Signavio, (n.d.). https://www.signavio.com/ (accessed August 31, 2018).

[50] Archi – Open Source ArchiMate Modelling, (n.d.). https://www.archimatetool.com/ (accessed September 7, 2018).

[51] K. Van Der Veken, Enterprise Architecture modelling to support collaboration - The ArchiMate language as a tool for communication, Master-Bu (2013). http://hdl.handle.net/1820/5019.

[52] R. Abraham, S. Aier, R. Winter, Fail Early, Fail Often: Towards Coherent Feedback Loops in Design Science Research Evaluation, in: Icis, 2014: pp. 1–12.

[53] E. Kabaale, L. Wen, Z. Wang, T. Rout, An Axiom based Metamodel for Software Process Formalisation : An Ontology Approach, in: Softw. Process Improv. Capab. Determ. SPICE 2017, Springer, Cham, 2017.

[54] Y. Wand, R. Weber, On the ontological expressiveness of information systems analysis and design grammars, Inf. Syst. J. 3 (1993) 217–237. http://doi.wiley.com/10.1111/j.1365-2575.1993.tb00127.x (accessed August 23, 2018).

[55] B. M., Treatise on Basic Philosophy, Volume 4, Ontology II: A World of Systems, (1979). https://philpapers.org/rec/BUNTOB-7 (accessed August 23, 2018).

[56] Y. Wand, R. Weber, An Ontological Model of an Information System, IEEE Trans. Softw. Eng. 16 (1990) 1282–1292. https://ieeexplore.ieee.org/abstract/document/60316/.

[57] R. Weber, Ontological Foundations of Information Systems, Coopers & Lybrand, Melbourne, 1997.

[58] J. Recker, M. Indulska, M. Rosemann, P. Green, Do Process Modelling Techniques Get Better? A Comparative Ontological Analysis of BPMN, Inf. Syst. J. (2005) 1–10. http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.89.1751&rep=rep1&type=pdf.

[59] M. Rosemann, J. Recker, M. Indulska, P. Green, A study of the evolution of the

representational capabilities of process modeling grammars, Lect. Notes Comput. Sci. (Including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics). 4001 LNCS (2006) 447–461. doi:10.1007/11767138_30.

[60] P.F. Green, M. Rosemann, M. Indulska, Ontological evaluation of enterprise systems interoperability using ebXML, IEEE Trans. Knowl. Data Eng. 17 (2005) 713–725. doi:10.1109/TKDE.2005.79.

[61] A. Saghafi, Y. Wand, Do ontological guidelines improve understandability of conceptual models? A meta-analysis of empirical work, in: Proc. Annu. Hawaii Int. Conf. Syst. Sci., 2014: pp. 4609–4618. doi:10.1109/HICSS.2014.567.

[62] A. Wroblewski, A. Leitner, Between Scientific Standards and Claims to Efficiency: Expert Interviews in Programme Evaluation, in: Interviewing Expert., Palgrave Macmillan UK, London, 2009: pp. 235–251. doi:10.1057/9780230244276_12.

[63] O.C. Robinson, Sampling in Interview-Based Qualitative Research: A Theoretical and Practical Guide, Qual. Res. Psychol. 11 (2014) 25–41. doi:10.1080/14780887.2013.801543.

[64] P. Clarke, A.L.M. Calafat, D. Ekert, J.J. Ekstrom, T. Gornostaja, M. Jovanovic, J. Johansen, A. Mas, R. Messnarz, B.N. Villar, A. O'Connor, R. V O'Connor, M. Reiner, G. Sauberer, K.-D. Schmitz, M. Yilmaz, An Investigation of Software Development Process Terminology, in: Commun. Comput. Inf. Sci., 2016: pp. 295–301. doi:10.1007/978-3-642-30439-2.

[65] C. Gonzalez-Perez, B. Henderson-Sellers, T. McBride, G.C. Low, X. Larrucea, An Ontology for ISO Software Engineering Standards, Comput. Stand. Interfaces. 48 (2016) 112–123. doi:10.1016/j.csi.2016.04.007.