

A Comparative Analysis of Using the Capability Notion for Congruent Business and Information Systems Engineering

Jelena Zdravkovic^{1*}, Janis Stirna^{1*} and Jānis Grabis^{2*}

¹ Department of Computer and Systems Sciences, Stockholm University, Box 7003, Kista, 16407, Sweden

² Information Technology Institute, Riga Technical University, Kalku 1, Riga, LV-1658, Latvia

jelenaz@dsv.su.se (orcid.org/0000-0002-0870-0330), js@dsv.su.se (orcid.org/0000-0002-3669-832X), grabis@rtu.lv (orcid.org/0000-0003-2196-0214)

Abstract. The notion of capability has been gaining a growing attention in the business and information system (IS) engineering community due to a number of reasons: it facilitates focus on business investments, it can be used as a baseline for business planning, and it directly leads to service specification and design. It is not however widely known to what extent capability is considered in different modeling approaches, how it is defined, and what purpose it fulfills. This article analyzes how the notion of capability is included in the frameworks spanning from business-oriented such as Business Architecture and Business Value Modeling, to the alignment-oriented represented by Enterprise Architecture (EA), and Enterprise Modeling (EM). The results of the analysis have shown that capability has widespread presence in the frameworks and that its conceptual meaning is largely similar, while the intentions and the mechanisms of its use differ, which raises stimulating opportunities for new contributions and improvements in the field.

Keywords: Capability, business modeling, enterprise architecture, enterprise modeling, information system development.

1 Introduction

Contemporary information system (IS) development frameworks and methodologies face complex challenges for being able to support organizations acting in highly competitive environments, addressing unexpected events, such as rapidly increasing customer demands, new legislations, new customers, and emerging alliances. Modern companies need to be sustainable in the presence of dynamically changing business conditions [1], [2], which is a business challenge as much as it is an IS development and management challenge. From a technical

* Corresponding author

© 2017 Jelena Zdravkovic et al. This is an open access article licensed under the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/4.0>).

Reference: J. Zdravkovic, J. Stirna and J. Grabis, “A Comparative Analysis of Using the Capability Notion for Congruent Business and Information Systems Engineering,” *Complex Systems Informatics and Modeling Quarterly, CSIMQ*, no. 10, pp. 1–20, 2017. Available: <https://doi.org/10.7250/csimq.2017-10.01>

perspective, the gap between business requirements and supporting IS is still present, mostly due to the fact that current IS development approaches, dominantly model-driven, operate with artifacts defined on a relatively low abstraction level. To go beyond the state of the art, development frameworks used by enterprises has to be structured for solving emerging problems, and enterprises have to have efficient methods for the use of these frameworks to deliver the right IS solutions just-in-time and just-enough. It is for this reason that ensuring the suitability of current IS development frameworks for supporting today's business structures and dynamics, is becoming increasingly relevant.

The notion of capability has a growing instance in the current business and IT alignment as well as development frameworks, for example, those that are business-oriented such as Business Architecture [3] and Business Value Modeling [4], those that focus on the business and IT such as in the area of Enterprise Architecture (EA) [5], [6], [7], [8], [9], and Enterprise Modeling (EM) [10], [11], [12], [13], [14], [15], [16].

Linguistically, *capability* means the ability or qualities necessary to do something [17]. The notion emerged in the beginning of the previous century in the context of competence-based management, military frameworks, and developing organization's competitive advantage [18], [19]. It was later considered for Business-IT alignment [20]. In brief, the emergence of the use of the notion of capability seems having the following motivations:

- (a) In the context of business planning, capability is becoming recognized as a fundamental component to describe what a core business does and, in particular, as an ability for delivering value beneath the business strategy;
- (b) In system development, capability makes IS designs (models) more understandable to business stakeholders by offering a notion for describing their IS requirements in the form that allows continuous model use by business and IT developers; and
- (c) Capability supports configurability of operations on a higher level of abstraction than what can be achieved by using only concepts such as services, business processes, and technology solutions.

Following these principles, capability has been conceptualized and included in a wide variety of frameworks such as for Business Architecture, Business Value Modeling, Enterprise Architecture, Enterprise Modeling, and, while there are clearly identifiable similarities, there are also differences in its use. To this end, the objective of the article is *to analyze the presence of capability in the current frameworks in a uniform manner*. The following aspects are analyzed: how capability is defined in a framework, which perspective it covers, how its development is supported, and what purpose it serves. The aim of this analysis is to obtain a systematic awareness of the presence and use of the notion of capability, to emphasize its role and importance in each considered framework, as well as to reveal similarities and differences in how capability is viewed by the frameworks. By analyzing these questions, we contribute towards a consensus about the notion of capability because a common understanding and use of concepts is among the key factors that positively influence successful adoption and standardization of IS development approaches.

The remainder of the article is organized as follows. Section 2 presents key requirements for IS development approaches based on the notion of capability in order to provide motivation for this research. Section 3 presents the frameworks analyzed. The frameworks are analyzed in Section 4. Section 5 summarizes the findings in terms or commonalities and differences. Conclusions and issues for future work are presented in Section 6.

2 Motivation and Requirements for Capability

The growing interest regarding capability, a concept for business and IS development, is motivated by the industry needs to provide new digital services tailored to specific customer requirements and to capitalize on organizations' core competencies. The industry requirements were elicited during interviews with three companies working in different sectors as part of an

EU financed research project CaaS – “Capability as a Service in digital enterprises”[†]. All three organizations had not previously used the capability concept as such but their businesses are based on continuous delivery of services and solutions in various situations. The solutions that were used were based on process variants and services supported by considerable efforts for manual customization and maintenance. In the beginning of the project, they were not accustomed to development approaches that offer explicit solutions for context based customization of business services and the notion of capability. In the early stages of explicating requirements for the CaaS solution, their business needs were discussed in the context of the current development approaches and the potential benefits of the capability concept. The requirements are documented in [21].

The first company is in the software development and service business sector specializing in solutions for utilities. Its main business is management of a variety of services for different customers and automation of exception handling processes. The second company is a business and IT consulting company specializing in providing e-government services. Within the scope of this investigation, its main concerns are customization of e-services for the needs of different municipalities and governance and promotion of these services taking into account specific circumstances of the municipalities. The third company provides cloud-based software development and maintenance services that enable companies to collaborate more effectively and demonstrate their regulatory compliance. Its business is based on a model driven software development tool.

The requirements elicitation objectives were (1) to identify digital services these companies are willing to improve and (2) to understand their expectations towards the improvement process. The first objective is beyond the scope of this article while findings concerning the second objective illustrate the industry expectations for using the capability concept.

The requirements elicitation process was organized as a series of face-to-face interviews with experts from each company. The interview findings were further refined in several collaborative modeling workshops inspired by the 4EM approach [22]. The main purpose of these workshops was the identification of goals that the companies have for the improvement of their digital services. The resulting integrated goal model was then refined together with representatives of the three companies. The goal model is also reported in [23].

The consensus among the companies was that they need to address the increasing complexity and variety of their services. Goals identified during the requirements analysis are attributed to the following four groups (see Figure 1):

- (1) *Organization specific business improvement goals*, such as some of the supporting goals of G1, i.e. G16 and G17 are beyond the scope of this article as they emphasize the overall needs for development of new business services or for improving the existing services for regulatory compliances.
- (2) *Digital service design (or development) goals* reflect the need to design capabilities providing the basis for digital services. The *capability design* (G2) is about a deliberate process of constructing capabilities and their supporting digital services. It should be guided by appropriate capability metrics (G5) and made efficient by reuse (G7). The goals formulated by the companies indicate that there should be means for analyzing capability designs. That includes both evaluation of the capability (G6) and customization of the design to meet specific needs depending on the context (G8).
- (3) *Digital service delivery (or run-time) goals* address the need for continuous monitoring, adjustment, and improvement of the capability once it is deployed. The adjustment implies that some aspects of the capability delivery are altered in response to changing business environment and operational performance without redesign and redeployment. This is represented by the third group of the goals concerning the *capability delivery* or run-time. The companies have needs for monitoring their operating environment and circumstances

[†] FP7 Project no. 611351 CaaS – “Capability as a Service in digital enterprises”, <http://caas-project.eu>.

(G9), as well as to evaluate actual performance (G10). The adjustments are made to deal with new circumstances and to improve the performance (G3). The companies also emphasize the resource allocation dimension of the capability delivery (G11). This goal implies that a capability cannot be successfully provided without having appropriate resources in sufficient quantity and that the resource provisioning is one of the main ways of adjusting capability delivery.

The need for capability analysis, resource allocation, and delivery adjustment naturally leads to the need for formalized representation of the capabilities which should be achieved by using appropriate modeling methods.

- (4) *Methodological support goals.* The capability design and delivery constitute a complex process. Hence, the companies expressed a need for methodological guidance supporting this process (G4). Capability design and delivery is perceived as a part of enterprise architecture and IT management (G12 and G13, respectively). Therefore, EA management frameworks are the prime candidates for formalized representation of capabilities. The capabilities should be developed and managed throughout their life cycles in an integrated manner using appropriate *capability management* solutions (G14), and the model-driven approach also contributes significantly towards achieving this goal.

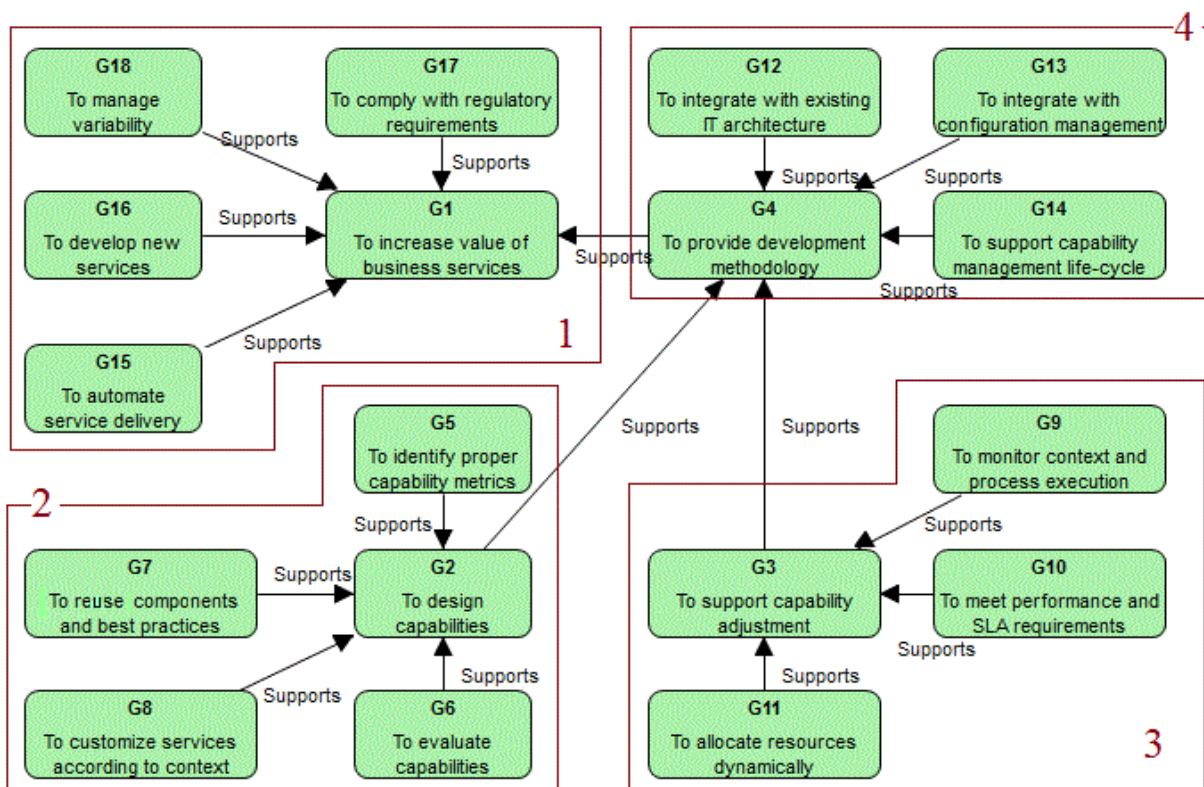


Figure 1. Goal model showing the need for approaches based on capability

In summary, the requirement analysis of the CaaS project suggested that there is a strong need for an approach that supports design and delivery of business in congruence with IT. This was addressed by developing an integrated methodology and development environment under the name of Capability Driven Development (CDD) [16], [24]. In the light of the concept of capability being used in a number of other approaches some of which are attempts at standardization, it became of interest to analyze how it is being addressed in these approaches and frameworks.

3 Background to Approaches Using Capability

3.1 Relevant Frameworks

The goal model (Figure 1) suggests that the key requirements towards modeling frameworks supporting the goals for a capability development methodology are as follows:

- The purpose of the methodology should be the development of digital services aligned with business strategy and implemented using information systems;
- Capability should be described (e.g., modeled) using a well-defined set of concepts and analyzed to support its delivery and evolution;
- A methodological support should be provided for capability design and delivery.

These requirements relate to conceptual, modeling, and process characteristics evaluated for different enterprise architecture management frameworks presented in [25]. The frameworks can be further classified according to their focus area of concern as: Business Architecture, Business Value Modeling, Enterprise Architecture, and Enterprise Modeling. The frameworks presented in Table 1 intend to fulfill the requirements summarized above by including capability-orientation in an integrated way, i.e. they aim to support all three steps, namely, design, delivery, and management of capabilities.

Table 1. The list of enterprise management frameworks included in the analysis

Focus Area	Framework
Business Architecture	BA (Business Architecture) – OMG [3]
Business Value Modeling	VDML (Value Delivery Modeling Language) – OMG [4]
Enterprise Architecture	NAF (NATO Architecture Framework) – NATO [9]
	DODAF (Department of Defense Architecture Framework) – US Department of Defense [7]
	MODAF (Ministry of Defense Architecture Framework) – UK Ministry of Defense [8]
	TOGAF (The Open Group Architecture Framework) – The Open Group [5]
	Archimate (EA modeling language) – The Open Group [6]
Enterprise Modeling	SOA (Service-oriented Architecture) – OASIS [10], The Open Group [11] and OMG [12] frameworks
	i* (pron. “i star”) – University of Toronto [13], [14], [15]
	CDD (Capability Driven Development) – FP7/Capability as a Service project, CaaS [16]

There are several other architectural frameworks, which are not considered in this article because they do not include the capability notion. Most notably the Zachman Framework [26] neither explicitly considers the capability notion nor it supports a model-driven approach to building IS aligned with the business strategy. Federal Enterprise Architecture Agency (FEA) [27] mentions capability as a mechanism for modularizing the requirements so that capabilities can be combined in new ways to achieve new business and technical objectives, but any formalism or way of using is not considered.

There are several other frameworks addressing the capability concepts. Scaled Agile Framework [28] provides patterns for development of enterprise-scale software applications and defines capability as a higher-level behavior of a solution. It has a clear purpose and methodological guidance although it does not cover representation and modeling aspects. The eSourcing Capability Model [29] describes best practices used in the sourcing domain. Its purpose is knowledge representation and analysis of capabilities along a number of dimensions, but it does not address the methodological issues. Capability Maturity Model [30] while

employing the concept of capability as such is aimed at improvement of software development processes of an organization instead of development and configuration of software applications.

To summarize, the analysis presented in this article is restricted to the frameworks having system development as a purpose and providing means and methodological support for modeling.

3.2 Business Architecture and Business Value Modeling

Business architecture focuses on the essence and structure of a business. OMG's proposal for *Business Architecture (BA)* is an enterprise blueprint aiming to provide a common understanding of an organization, as well as to align strategic objectives with tactical demands [3]. The major building blocks of this architecture are high-level domains representing different *aspects* of a business, such as: Vision, Strategies, Policies, Regulations (“why”); Organization, Stakeholders (“who”); Products, Services, Information, Capability (“what”); Value Streams (“how”); and Metrics&Measures (“how well”). The architecture proposal has its origin in the work [31] which identified the need for creating a Capability Map of an organization and using it as a foundation for Business/IT alignment.

Value Delivery Modeling Language (VDML) [4] provided by OMG, defines a modeling language for analysis and design of the operations of an enterprise with a focus on the creation and exchange of value. Its aim is to provide an abstraction of the operations appropriate for business planners. VDML links strategy and business models to the activities, roles, and capabilities that run the enterprise.

3.3 Enterprise Architecture

EA is practice of capturing, documenting, and analyzing enterprise designs using a holistic approach typically focusing on a number of perspectives (e.g. business, information, process, technology). A number of EA frameworks have been created for various purposes and domains. Some of them mostly focus on defining the way EA should be documented while others also prescribe methodologies for EA development. This section presents some of the more widely used frameworks that include the concept of capability, namely, TOGAF, ArchiMate, DODAF, MODAF, and NAF.

TOGAF. *TOGAF* describes architecture domains, concepts and methods for designing, using, and maintaining an EA [5]. The Business Architecture *domain* defines strategy, governance, organization, key business processes, and, here, the notion of capability is considered. The Architecture Development Method (ADM) describes a phase-based process for developing architecture domains, including establishing an architecture vision and framework, developing architecture content, transitioning (implementation/migration), and governing the realization of architectures.

ArchiMate. *ArchiMate* is an EA modeling language [6]. It provides an architectural approach to describe and visualize different concepts of the types: active structure, behavior, and objects, as well as it defines their relations. Based on specializations of the core concepts, the language defines three main *layers*: Business, Application and Technology. The structure of ArchiMate closely corresponds to the concepts of TOGAF, and complements it with aspects such as a graphical representation for creating an integrated architecture model in the form of different viewpoints. Each of the viewpoints presents a different perspective on modeling providing the motivation that underlies the Enterprise Architecture and allows a modeler to focus on certain aspects.

Architecture Frameworks Originating from Military Applications. Several architecture frameworks have originated from the military domain. We have analyzed the three most widely used – DODAF, MODAF, and NAF.

DODAF – The Department of Defense Architecture Framework [7] is an architecture framework for the US Department of Defense that provides visualization infrastructure for specific stakeholder concerns organized by various *viewpoints*, such as the following: All Viewpoint, Capability Viewpoint, Data and Information Viewpoint, Operational Viewpoint, Project Viewpoint, Services Viewpoint, Standards Viewpoint, and Systems Viewpoint.

MODAF – The UK Ministry of Defence Architecture Framework [8] is an architecture framework that defines a standardized way of conducting EA. It is originally developed by the UK Ministry of Defence. It is organized according to the following views: Strategic, Services, Logical, Physical, Programmatic, Standards, and All Views. It is also used outside UK, e.g. by Swedish Armed Forces [32].

NAF – The NATO Architecture Framework [9] is an EA framework developed by NATO; it has many commonalities with DODAF and MODAF. A key difference is that MODAF is a description framework, i.e. it does not have its own methodology, while the latest version of NAF has a methodology based on TOGAF. NAF has the following views – All Views, Capability View, Operational View, System View, Service View, Technical View, and Programme View.

3.4 Service Oriented Architecture

Service Oriented Architecture (SOA) is the paradigm for creating EA that is based upon the use of services – stand-alone units of functionality available to their consumers via a formally defined interface. A key to SOA is that business interactions occur with services operating independently. Owing to its core principles fostering development of autonomous and loosely-coupled services and in return enabling their reuse, as well as orchestration to larger services, SOA is seen to provide both time-to-market advantages and business agility.

The specifications for SOA are provided by three main sources: *OASIS* – an abstract, foundation reference architecture addressing the *ecosystem viewpoint* for building and interacting within the SOA paradigm [10]; *The Open Group* – a layered reference architecture from a consumer and provider perspective with cross-cutting concerns describing architecture building blocks and principles that support the realizations of SOA [11]; and *SOA Modeling Language (SoaML)* by OMG – defining a small set of extensions to UML to support SOA modeling [12] (it can be seen as an instantiation of a subset of The Open Group’s architecture for representing SOA artifacts in UML).

3.5 Enterprise Modeling

EM is a process where an integrated and negotiated model describing different aspects of an enterprise is created for the purposes of (1) developing the business, (2) ensuring the quality of business operations, and (3) using EM as a problem-solving tool. The typical aspects modeled are business goals, process, concepts, actors, as well as IS requirements. Many EM approaches and tools have been developed and used in practice. This article considers two of them: *i** and Capability Driven Development that lately have addressed capability modeling.

*i** is an approach for EM and requirements engineering to capture intentions and related business actors, and to operationalize business goals through concrete actions and needed resources [13]. *i** facilitates an exploration of enterprises by providing a graphical depiction of actors, their intentions, dependencies, responsibilities, and alternatives. Actors including Agents and Roles and associations between them represent the social aspect of *i**. Actors’ intentions are expressed within the actor’s boundary using Goals and Soft-goals, Tasks to be performed to achieve the goals, and needed Resources. The framework enables creating two types of models: (a) Strategic Dependency Model (SDM), which is used to express the network of intentional, strategic relationships among actors dependent on each other to accomplish tasks, provide resources, and satisfy goals and soft-goals; and (b) Strategic Rationale Model (SRM) which

describes each of the networked actors in detail – internal goals, resources, tasks, etc. Capability has been recently included into i* [14], [15].

CDD – Capability Driven Development [16] is a methodology developed by the CaaS project to support continuous delivery of value and take advantage of changes in business context and technologies. CDD consists of a modeling language and notation, as well as detailed guidelines for the way of working covering three key aspects of capability life-cycle in organizations, namely: (1) capability design, including EM; (2) capability delivery, including adjustments to context changes; and (3) feedback-loop of lessons learned for continuous business improvement. CDD is supported by a modeling tool, an application for monitoring capability delivery, and a context monitoring platform.

4 Analysis

This section analyses the approaches outlined in Section 3 according to the following aspects that are relevant for understanding and choosing a modeling method or its components for the use in practice.

Formalism addressing the issue of how capability is to be documented i.e. semantics and notation. In the context of this analysis we have investigated – what is the capability definition and how capability is represented in the meta-model in terms of key relationships with other modeling components. Concerning the notation for representing capability the approaches included in this analysis are all based on the principles of Conceptual Modeling and hence they have defined modeling notations. Analyzing the specifics of the different modeling notations is besides the main thrust of this article.

Methodology associated with a modeling framework should describe in concrete terms how to elicit and document the relevant concepts in the framework according to the formalism. It may also cover prerequisites and resources needed. The procedure should be described in terms of work steps to be performed with input, output, and tool support. In some cases, the procedure also includes guidelines of modeling. The level of detail to which developers of modeling frameworks and approaches chose to elaborate modeling methodologies varies in practice. This analysis has attempted to analyze, in broad terms, what ways of working are defined for capability modeling.

Perspective refers to the modeling approaches, frameworks, and methods often addressing several views of the modeling domain, for instance, goals, business processes, actors. They are often called perspectives, views, or viewpoints if they are defined as part of the development methodology. The perspectives urge the modeler to look at the enterprise from a specific angle and guide the modeling process in a way that specifically captures and analyzes the specific perspective. The perspectives facilitate the modelers' focus on a specific aspect of the domain at a time while at the same time offering possibilities for integrating with other perspectives thus providing a more holistic, multi-perspective view on the organization. In essence, perspective influences what is considered important when following the modeling procedure defined by the method. More information on the notion of perspective in EM is available in [22]. In the context of this analysis we have focused on what views or perspectives does the approach offer for capability modeling.

Purpose addresses the rationale for using a particular method component for representing a particular aspect of the modeling domain. This analysis has been concerned with what part of the modeling domain is represented by the notion of capability.

4.1 BA-OMG

Formalism: Business capability describes what a business does – specifically, it is an ability or capacity that the business may possess or exchange to achieve certain outcome. In BA the four concepts are fundamental, stable and related to each other: Value (Stream), Organization,

Information, and Capability, where according to the meta-model Capability is delivered by Organization and supported by Information to enable a Value Stream stage. These core concepts further extend to include Strategy, Stakeholders, Products, Services, Policies and other concepts, which are more dynamic in the nature and more often changed (removed, extended, etc.). E.g., if the business wants to introduce a new product, the product planning must be aligned with the Value Stream and Capability. The meta-model for the architecture has not yet been completed.

Perspective: BA defines the capability map as the viewpoint of what a complete business does, i.e. its basic building blocks.

Methodology: Capabilities are elicited and composed to deliver strategic actions in the Value Stream of an organization. A capability is elicited by including the resources from business units' components (such as skills and objects), and when needed, these components are altered, removed, or added, while the overall configuration of the Capability Map is kept according to the Value Stream. Furthermore, a capability can be decomposed to sub-capabilities by asking "what", e.g., Partner Management capability can have sub-capabilities: Partner Preference Management and Partner Matching. There is, however, no detailed identification method available.

Purpose: Capabilities provide a common and complete vocabulary ("map") for what of the business/organization. By being able to focus on concrete functional points of the business and ignoring other "noise", capabilities enable business and investment analysis.

4.2 VDML

Formalism: A Capability represents the ability of an organization to perform a particular type of work and may involve people with particular skills and knowledge, intellectual property, defined practices, operating facilities, tools, and equipment.

The VDML defines the following key relationships for capability:

- A capability may consist of several capabilities.
- A capability may be performed by one actor or a team of actors.
- A capability may rely on other capabilities to perform specific activities.
- Capabilities typically are sharable in different contexts to achieve economies of scale or consistent controls for multiple uses.
- Capabilities are owned by organization units.
- Capabilities are delivered by capability offers, for instance, when a capability needs to be delivered in changing context.
- Capability offers are described by capability methods each of which has a set of measurements and assignments of performers.
- A capability method is associated with a capability offer owned by the organization unit that has resources to provide the capability.
- Capabilities apply different, repeatable activity patterns to support each capability offer.

Perspective: VDML supports a Capability Viewpoint for specification of capabilities that the organization has, i.e. creating overviews of the existing capabilities and capability heat-maps.

Methodology: VDML documentation describes a high-level way of working but does not provide detailed modeling guidelines.

Purpose: The purpose is to create capability taxonomy, to promote consistency in the definitions of similar capabilities, and to support the analysis of the same or similar capabilities of different organizations. This allows realizing economies of scale as well as provides reference to existing capabilities, e.g. when a capability is needed to respond to changing markets or to develop a new product or line of business.

4.3 TOGAF

Formalism: Capability is an ability expressed in high-level terms including a combination of organization, people, processes, and technology to achieve it. It is “a business-focused outcome that is delivered by the completion of one or more work packages. Using a capability-based planning approach, change activities can be sequenced and grouped in order to provide continuous and incremental business value”. In both core and full meta-models of the current TOGAF (version 9.1):

- Function entity is used to describe and deliver a unit of business capability.
- Function is bounded by business services that explicitly define interfaces to capabilities usage.
- In the meta-model “extension” for the infrastructure consolidation, Capability is represented by its own entity with the attributes: Business value and Increments.

Perspective: Capability focuses on business outcomes (functions), as well as it addresses coordination of projects across corporate functional domains that together contribute to a capability. It is addressed in the Architecture Principles, Requirements, and Roadmap perspective.

Methodology: Capability-based planning focuses on the top-down planning, engineering, and delivery of strategy-based business capabilities to the enterprise. It follows ADM, i.e. when defining Architecture Vision a corporate strategic plan by goals and objectives is created; it is used to develop Architecture Definition by defining capabilities as business outcomes and refining them using business scenarios. Transition Architecture phase focuses on Capability Increments creating discrete deliverables delivered and coordinated by the Implementation and Migration Plans.

Purpose: Using capability-based planning, the key is that all the architecture will be expressed in terms of business outcomes and values rather than in IT terms, thereby ensuring IT alignment with the business. Capability-based planning frames all phases of the architecture development in the context of business outcomes, clearly linking the IT vision, architectures, and the implementation and migration plans with the corporate strategic, business, and line of business plans. It also copes well with the friction of coordinating projects across corporate functional domains that together enable the enterprise to achieve that capability (e.g. “electronic service delivery”).

4.4 ArchiMate

Formalism: A capability represents an ability that an active structure element, such as an organization, person, or system, possesses. Capability is a behavior element in the generic meta-model which is a main inheritance-based hierarchy including the behavior and structure elements of the language [6].

Perspective: The capability map viewpoint provides an overview of the capabilities of the enterprise.

Methodology: the language does not include any specific methodology for its use.

Purpose: A capability shows specific outcomes delivered by an enterprise. When viewed together, capabilities can be used as a heat map to identify areas of investment.

4.5 DODAF

Formalism: Capability is defined as the ability to achieve a Desired Effect under specified (performance) standards and *conditions* through combinations of ways and means (activities and resources) to perform a set of activities [7]. Condition means the state of an environment or

situation in which a Performer performs; Desired effect – desired state of a resource; Resource – data, information, performers, materiel, or personnel types that are produced or consumed.

DODAF meta-model defines the following key relationships of capability:

- Capability is related to ways and means in terms of resources and activities.
- Desired effect is a desired state of a resource defined by a goal and a capability determining states (persistence of current ones, or changes to future ones).
- Capability is associated to measures through the activities they entail and the desired effects sought.
- Capabilities relate to services via the realization of the capability and performer association denoting a service. According to DODAF, a Service would not provide the desired effect(s) but, rather, access to ways and means (activities and resources) that would.

The originators of DODAF have created a modeling notation, but its implementation in modeling tools varies to some extent.

Perspective: Capability is part of the Capability Viewpoint. It is organized in the following models each of which focus on a specific aspect of capability design:

CV-1: Vision – used for describing project’s vision, goals, objectives, plans, activities, events, conditions, measures, effects, and produced objects.

CV-2: Capability Taxonomy – used to define a data repository of all terms used on an architectural level.

CV-3: Capability Phasing – used to specify the capability design at different points in time or during specific periods of time. Activities, conditions, desired effects, rules complied with, resource consumption and production, and measures are specified independently of the performer and location solutions.

CV-4: Capability Dependencies – used to specify dependencies among and grouping of capabilities.

CV-5: Capability to Organizational Development Mapping – used to model the planned capability deployment solution and interconnection for a particular capability phase, including definitions of performers, locations, and their associated concepts.

CV-6: Capability to Operational Activities Mapping – used to align the capabilities required and the activities that those capabilities support.

CV-7: Capability to Services Mapping – used to map the capabilities and the services that these capabilities enable.

Methodology: DODAF explains the overall purpose and structure of the viewpoints and modeling constructs, but it does not provide a detailed methodological guidance for eliciting, capturing, and representing the different modeling components.

Purpose: According to [7], the concept of capability supports capturing and analyzing these kinds of questions: how does a particular capability or capabilities support the overall mission/vision; what outcomes are expected to be achieved by a particular capability or set of capabilities; what services are required to support a capability; what is the functional scope and organizational span of a capability or set of capabilities; what is a current set of capabilities that is managed as part of a portfolio.

4.6 MODAF

Formalism: Capability definition is as follows: a high level specification of the enterprise’s ability. A capability is a classification of some ability – and can be specified regardless of whether the enterprise is currently able to achieve it. If a capability cannot be achieved it can still be planned to be achieved in the future. Capabilities in MODAF are not time-dependent – once defined they are persistent. MODAF allows the architects to develop a formal taxonomy of capabilities which can be re-used across multiple architectures [8], and it has a well elaborated

and formally defined meta-model. The key relationships of capability are the following:

- Capability supports an enduring task. In this case enduring task is seen as an operationalization of vision and enterprise's goals; in other frameworks capability is directly linked to goals.
- Capability has associations with other capabilities of type "depends on", "contains", and "specializes".
- Capability is constrained by environment.
- Capability has measurable property.
- Capability is supported by standard operational activity.
- Capability is delivered by capability configuration.
- Service aims to achieve capability.

The modeling notation used is based on UML.

Perspective: Capability is defined in the Strategic Viewpoint and further operationalized in Services Viewpoint.

Methodology: MODAF users have defined the overall development process such the MODAF Architecting Process [8] or an approach to model-based capability development developed for the Swedish Armed Forces. These methodologies explain the overall purpose and structure of the viewpoints and modeling constructs, but they do not offer detailed guidance for eliciting, capturing, and representing the different modeling components of the framework.

Purpose: According to MODAF, capabilities are high-level specification of the enterprise's ability addressing the question of what capabilities are required for a given scenario or operation. Capabilities are to form a capability taxonomy that is seen as one of the most important parts of a MODAF architecture.

4.7 NAF

Formalism: Capability is defined as "the ability of one or more resources to deliver a specified type of effect or a specified course of action". Capability is included in the NAF meta-model [9]; its interactive website also has the ability to present NAF according to MODAF views. The key relationships of capability are:

- Capabilities may be specialized into more specific capabilities, composed of several capabilities, and dependent on other capabilities.
- Capability when applied is associated with measurable categories.
- Capability is elaborated into Capability configuration package, which is used to configure resources for capability implementation.
- Enterprise phase exhibits a capability. The connection between capabilities and goals is realized through enduring phase of the enterprise.
- Capability supports an enduring task by defining capability for task.

Perspective: Capability view, further specified in models addressing detailed aspects of capability development, namely: Capability vision, Capability taxonomy, Capability phasing, Capability dependencies, Capability to organizational deployment, Operational activity to capability mapping.

Methodology: The latest version of NAF has a methodology and it is under further development. It integrates TOGAF's ADM, components of MODAF, and systems engineering standards such as ISO15288. There are guidelines for tool usage for example for IBM Rational System Architect.

Purpose: Capability is used for the specification of an ability to achieve an outcome.

4.8 SOA

Formalism: OASIS: a capability is a business functionality of a service provider that addresses a well-defined need of service consumer. In the meta-model, Capability is the entity being offered by a Service Provider to satisfy the Need of a Consumer. *The Open Group:* a capability represents a category of requirements that fulfills a strongly cohesive set of needs (functionality). In the meta-model, Capability is a part of Architecture Building Block (a constituent of the architecture model). *OMG/SoaML:* a capability specifies a cohesive set of functions or resources that a service provided by one or more participants might offer. In the meta-model, Capability realizes a Service Interface.

Perspective: OASIS: a capability is a part of the Visibility Model, a viewpoint elaborating the relationships between the participants (human social structures, such as service consumer, and automated IT mechanisms, such as service interface) interacting with each other in the context of a SOA. *The Open Group:* The SOA Reference Architecture is partitioned into nine layers (Consumer, Business Process, Service Component, etc.), each being a viewpoint on a specific subset of characteristics, and responsibilities that relate to unique value propositions within a SOA. A layer consists of a set of Architecture Building Blocks providing the key responsibilities to its layer, and each block is supported by a set of capabilities. *OMG/SoaML:* there is no specific perspective based on capability, or containing capability.

Methodology: OASIS and The Open Group provide the description and the overall purpose and structure of capability, but they do not define detailed methodological guidance for identifying and representing capability. *OMG/SoaML* describes a brief methodology that suggests modeling a business process collaboration as the starting point on the basis of which each participant and its activities' service interfaces are elicited and realized by capabilities. Another methodology for capability identification is elaborated by IBM RUP for SOMA [33] describing three main techniques: (1) Goal-service modeling, which identifies capabilities needed to realize business requirements such as strategies and goals; (2) Domain decomposition, which uses activities in business processes and other descriptions of business functions to identify needed capabilities; and (3) Existing asset analysis, which mines capabilities from existing applications.

Purpose: OASIS: a capability is a real world effect that a service provider is able to provide to fulfill the need (request) of service consumer. *The Open-Group:* a capability allows focusing on the "what" rather than the "how"; by its aggregation to an ABB, a capability is also related to the implementation part of the block, i.e. its application solutions and used technologies. This enables business capabilities to be aligned to the technical capabilities required to service them. *OMG/SoaML:* capabilities fulfill business requirements by defining the corresponding functionality behind the interfaces of services.

4.9 i*

Formalism: Capability is an intentional autonomous bundle of organizational resources that are built and evolved over time. Capabilities are modeled as specialized Actors (Roles), meaning a position within the enterprise responsible for a capability (in case of collaborating partners the position can be in a partner enterprise), and possibly have dependency association with another capability.

Perspective: Capabilities are enterprise-wide; they embody knowledge and skills of people, business processes, and technical systems. The aim of the approach is to respond to the dynamic aspect of strategic business requirements, hence capabilities should be developed, orchestrated, and deployed to provide different choices (alternatives) of knowledge and skills for supporting these requirements. The SDM perspective is used to depict the overall capabilities' configurations and relationships. The SRM perspective expresses alternative means of achieving individual capability's goals and their contribution to quality attributes represented as soft-goals by specifying required tasks and resources.

Methodology: The main method describes three steps for designing capabilities. Step 1 – *capability development alternatives:* the context of the organization is depicted with capabilities using SDM. Further, each capability is using SRM elaborated for alternatives (for example, the choice to either develop a sales system in-house, or to adopt a software-as-a-service solution). that is modeled by a SRM of the capability. Step 2 – *capability orchestration alternatives:* it is determined if a capability should be a part of another, or dependent on another capability for its completion. Step 3 – *Deployment configurations* – how to configure a designed capability at the deployment time [14].

Another method is for increasing flexibility when aligning strategic management and IT systems architectures. Using a dependency propagation graph, the method examines (1) how tightly capabilities are coupled to other capabilities and organizational entities and (2) how and what kinds of impacts (strategic intent) to expect from a change in a capability throughout the organization; e.g., if many other capabilities depend on a capability, and SOA itself is highly ranked because of a high effort required for its engineering, then this leads to inflexibility [15].

Purpose: Capabilities bundle the realization alternatives of strategic business requirements by including different sets of knowledge and skills starting from goals and towards tasks and resources. Capabilities can be combined and complemented to achieve composite capabilities. Modeling capabilities as i* actors allows to analyze how different capabilities relate to one another and to facilitate reasoning on (IT) capability alternatives.

4.10 CDD

Formalism: Capability is the ability and capacity that enable an enterprise to achieve a business goal in a certain context. It is defined according to the capability meta-model [16]. In the meta-model capability has the following key relationships:

- Capability is required by goal. A capability supports fulfillment of a goal, i.e. the goal needs to be satisfied during the capability delivery.
- Capability influences KPI and during capability delivery KPIs are measured.
- Capability requires process. Capability requires one or several business processes, i.e., a capability is delivered by executing business processes.
- Capability requires context set. A capability is designed to be used in a certain context situation that is represented by a context set. Context indicators are measured in the delivery phase.
- Capability is supported by pattern. The pattern component describes the actual solution for realizing a capability in the given context situation (represented by context set at design time) and by context indicators during run time. Patterns are linked to adjustment algorithms – expressions for context and KPI calculations that trigger run-time adjustments of capabilities.

The CDD approach also specifies a notation for modeling, although it is flexible and may be adapted or replaced entirely depending on the tooling used.

Perspective: Capability Design view which supports modeling goals, KPIs, the context in which the capability is relevant, and the associated variations of the business process used to deliver the capability.

Methodology: The CDD methodology provides detailed guidelines for designing capabilities and is supported by a Capability Design Tool.

Purpose: Capability represents how organization's business value is delivered depending on changes in the application context.

5 Analysis and Discussion

The capability concept has a significant presence in the frameworks that we have analyzed. All of the frameworks use models to represent organizational designs. Yet they specify various alternatives for designing and utilizing a capability. In this section, for each criterion used in Section 4, we provide a summary in Tables 2–5, namely, formalism (Table 2), perspective (Table 3), methodology (Table 4), and purpose (Table 5), and discuss respective commonalities and differences of the frameworks.

Table 2. The “Formalism” aspect of the capability concept

	Formalism (Definition part)
OMG-BA	The ability or capacity that the business possesses or exchanges to achieve certain outcome.
OMG-VDML	The ability of an organization to perform a particular type of work; and may involve people with particular skills, intellectual property, defined practices, operating facilities, tools, and equipment.
TOGAF	The ability expressed including a combination of organization, people, processes, and technology to achieve it.
ArchiMate	An ability that an active structure element, such as an organization, person, or system, possesses.
DODAF	The ability to achieve a Desired Effect under specified standards and conditions through combinations of means to perform a set of activities.
MODAF	A high level of enterprise’s ability specified regardless of whether the enterprise is currently able to achieve it.
NAF	The ability of one or more resources to deliver a specified type of effect or a specified course of action.
OASIS SOA	Ability to deliver a real world effect as a business function of a service provider that addresses a need of service consumer.
O. G. SOA	A category of requirements that fulfills a set of needs.
OMG SoaML	A set of functions or resources that a service provided might offer by one or more participants.
i*	An intentional autonomous bundle of organizational resources.
CDD	The ability and capacity that enable an enterprise to achieve a business goal in a certain context.

Commonalities in formalism. All the frameworks consider capability as the organization’s ability for delivering a business function. In some of them, such as in BA, VDML, DODAF, MODAF, NAF, and CDD, it is a key concept tying together organization’s intentions with operational implementations. In the others, capability has a supporting role to a representative framework’s component – such as in TOGAF to enable a Function, or in SOA to realize a Service. While all frameworks address the aspect of measurement, formalisms of some offer more explicit means for capability monitoring, for instance, VDML meta-model, called Structured Metrics Metamodel, includes constructs for measuring various aspects of the design; in MODAF and NAF the capability is associated with measurable categories; in CDD it is linked to context indicators and measurable properties.

Differences in formalism. Some frameworks integrate capability with IS solutions (technology), such as SOA and CDD, while the others do not do this explicitly and on a detailed level. Some frameworks offer explicit modeling components for modeling context (e.g. CDD has context element, measureable property, context situation), while NAF has an explicit focus on measuring according to environmental factors. Some others, such as BA, TOGAF, and i* do not have dedicated modeling constructs for context representation and hence modeling constructs such as measures, should be used.

Commonalities in perspective. The “integrational” nature of capability has been observed, i.e. it is used to bind the strategic/intentional part of the organizational design with the operational or

technical parts. Hence, capability should be seen as a key concept relevant to both business planners as well as operational planners. In VDML, DODAF, and NAF capability even has its own perspective. BA and ArchiMate support a collective capability perspective through maps. Others define specific perspectives including capability together with other modeling elements.

Table 3. The “Perspective” aspect of the capability concept

	Perspective in which capability is included
OMG-BA	Capability map
OMG-VDML	Capability viewpoint
TOGAF	Architecture principle, requirement, and roadmap perspective
ArchiMate	Capability map
DODAF	Capability viewpoint
MODAF	Strategic viewpoint
NAF	Capability view
OASIS SOA	Visibility model
O. G. SOA	Architecture layer
OMG SoaML	Not supported
i*	Strategic dependency model and strategic rational model perspectives
CDD	Capability design view

Differences in perspective. Some approaches, such as BA, VDML, TOGAF, DODAF, ArchiMate, and i* are catering for overseeing the capability map, while others, e.g. CDD, mostly focus on the specific capability delivery aspects. Also, some approaches such as NAF, i*, and CDD, address capability to support variability of the solutions as part of the capability perspective, while the others do not. Context related variability and its monitoring are addressed only in CDD and NAF.

Table 4. The “Methodology” aspect of the capability concept

	Methodology
OMG-BA	High level of functional description. No real modeling support.
OMG-VDML	High level of process description. No real modeling support.
TOGAF	Capability-based planning top-down method, which follows the phases of ADM.
ArchiMate	Not supported.
DODAF	Overall description, but no detailed guidance for modeling.
MODAF	Overall EA development processes defined specifically for UK Military of Defense and Swedish Armed Forces.
NAF	Based on TOGAF; in addition, local conditions are considered for capability design.
OASIS SOA	No explicit method.
O. G. SOA	No explicit method.
OMG SoaML	High-level description for design.
i*	High-level guidelines for development, orchestration, deployment, and management.
CDD	Methodology with guidelines for capability design and runtime adjustments, supported by a tool.

Commonalities in methodology. Majority of the frameworks currently do not provide a methodological approach for capability elicitation and development. The ones that provide a guidance do so on a fairly high level of granularity, i.e. provide suggestions on a large scale about, for instance, development phases, but not about detailed aspects of eliciting and analyzing

specific design constructs. Furthermore, the main thrust of the guidance focuses on design-time while guidance for run-time is assumed to be supported by other approaches.

Differences in methodology. The CDD framework follows Situational Method Engineering [34] and is structured in self-contained method components (“chunks”). It includes aspects of design and adjustments of capabilities at run-time based on local conditions. The NAF framework is the closest to CDD in this respect by considering local conditions in design of capabilities as well as in post-run-time measurements for monitoring capability delivery. NAF does not have a method for capability adjustments at run-time. The specifications of the other frameworks provide methods neither for use of capability at run-time, nor for adjustments.

Table 5. The “Purpose” aspect of the capability concept

	Purpose
OMG-BA	Capabilities provide a common and complete vocabulary for <i>what</i> of a business, and define discrete functional abilities.
OMG-VDML	Capability taxonomy promotes consistency in the definitions of similar capabilities, and to support the analysis of the same or similar capabilities of different organizations.
TOGAF	An architecture is expressed in terms of business outcomes (capabilities) and values, to ensure Business-IT alignment.
ArchiMate	Capabilities show specific outcomes or when viewed together, they can be used as a heat map to identify areas of investment.
DODAF	Define capability support for goals, outcomes, needed services, and map the overall capability portfolio.
MODAF	Capabilities enable creating a high-level specification of the enterprise’s ability.
NAF	Capability is used for the specification of an ability to achieve an outcome.
OASIS SOA	Capability is a real world effect that a service provider is able to provide to fulfill the need of a service consumer.
O. G. SOA	Capability enables focusing on the “what” rather than the “how”; it is aligned with the technological part of the capability.
OMG SoaML	Capabilities support business requirements by defining the functionality behind the interfaces of services.
i*	Capabilities bundle the realization alternatives of strategic business requirements, starting from goals and towards tasks and resources.
CDD	Capability models how firm’s business value is delivered depending on context changes.

Commonalities in purpose. The frameworks agree that capability facilitates the process of defining and bundling discrete functional abilities/outcomes of a company.

Differences in purpose. Frameworks focusing on IS modeling, such as SOA and CDD, consider using the capability concept to facilitate delivery of values or services of a single company. Frameworks that are mainly organization-oriented such as BA, VDML, TOGAF, DODAF, ArchiMate, and i* see the capability as a means of an easier integration with other companies and partners, by being able to show abilities by distinct functionalities or through a “capability map”. Another area of use is assessing potential synergies of, e.g. how a capability could span several projects, units, or even companies, i.e. analyzing the capability portfolio horizontally. Another difference is that some frameworks set capability as the key modeling construct and hence consider it important for investment or improvement analysis, while others, e.g. SOA, do not have this aim.

6 Concluding Remarks

In this article we have analyzed how the concept of capability is used in different frameworks for business analysis, enterprise architecture management, and enterprise modeling. While there are differences in the meta-models, all of the analyzed contributions use capability to bind the intentional part of the organizational design with the operational part that also encompasses

variability/alternatives. The exact details of this differ depending on the purpose of the framework but the main principles remain the same. In this respect we did not aim to create a common ontology for capability, but our findings could be used as a basis for such a task; or they could be linked with an existing attempt that presents an analysis of the ArchiMate's proposal for coverage of portfolio management including concepts such as capability and resource to create a unified ontology which integrates business strategy with portfolio management [35].

The motivation behind this study lies in the fact that for a new paradigm of business modeling and IS engineering to make an industrial impact, prevalence and standardization aspects in usage has to be considered. As for capability-oriented engineering, since its presence in modeling is recent, a relevant effort is to analyze its use in different framework proposals in order to a) align them by mapping of concepts, or b) complement concepts of a one framework by concepts of another, and thus enable a more transparent cross-framework use. Alignment between frameworks is desired due to their focus on different levels of design detail (strategic, operational), which means that in practice one may need to combine a strategic capability design expressed in one framework with an operational realization in another. For example, NAF or BA architecture design can be further elaborated and then executed with the support of CDD. The importance of this issue is further raised by the fact that some frameworks only offer tool support for design while others are also able to support run-time governance, monitoring, and adjustment. However, aligning frameworks that are of less natural fits, to one another could be a challenge and is among issues for future work.

Despite the fact that the differences in the definitions of capability are not being major in the analyzed frameworks, the link between the capability and the IT solutions is achieved to a different degree. This kind of relationship should be supported by modeling tools considerably more than it is done currently and not only documenting capability designs but also run-time monitoring and improvement. The latter aspect particularly stresses the need for new developments in the area of tools, because it requires model-level integration with applications for context modeling, pattern and model repositories, monitoring dashboards, ERP systems, etc. In essence, capability offers a construct that can be used for design and at run-time, but the current methods and tools seem to offer only a limited guidance and support for its usage at run-time.

Acknowledgments. This work has been partially supported by the EU-FP7 funded project no. 611351 CaaS – Capability as a Service in Digital Enterprises.

References

- [1] C. Lucke, S. Krell and U. Lechner, "Critical Issues in Enterprise Architecting – a Literature Review," AMCIS'2010, 2010. Available: <http://aisel.aisnet.org/amcis2010/305>
- [2] J. Stirna and J. Zdravkovic, "Interview With Sladjan Maras on Challenges and Needs in Enterprise Modeling," BISE Journal, Springer, vol. 57, no. 1, pp. 79–81, 2015. Available: <https://doi.org/10.1007/s12599-014-0358-9>
- [3] OMG Business Architecture Special Interest Group & Guild, "A Guide to Business Architecture of Body Knowledge (BIZBOK) 5.5," Jan. 2017 (available to members). Available: <http://www.businessarchitectureguild.org/?page=BIZ>
- [4] OMG, "Value Delivery Metamodel™ (VDML™) Version 1.0," 2013. Available: <http://www.omg.org/spec/VDML/1.0/>
- [5] The Open Group, "TOGAF Version 9.1, an Open Group Standard," The Open Group online. Available: <http://pubs.opengroup.org/architecture/togaf9-doc/arch/index.html>
- [6] The Open Group, "ArchiMate 3.0 Specification," Published by The Open Group, July 2016. Available: <http://www2.opengroup.org/ogsys/jsp/publications/PublicationDetails.jsp?catalogno=I162>
- [7] Department of Defense, "DoDAF Architecture Framework Version 2.02," 28 May 2009. Available: http://dodcio.defense.gov/Portals/0/Documents/DODAF/DoDAF_v2-02_web.pdf

- [8] UK Ministry of Defence, “MOD Architecture Framework,” 2012. Available: <https://www.gov.uk/guidance/mod-architecture-framework>
- [9] UK Ministry of Defence, “Proposed NAF v4 Meta-Model (MODEM),” NATO Architecture Framework v4.0 Documentation, 2013. Available: <http://nafdocs.org/modem>
- [10] OASIS, “Reference Architecture Foundation for Service Oriented Architecture Version 1.0 Committee Specification,” pp. 118, 4 Dec. 2012. Available: <http://docs.oasis-open.org/soa-rm/soa-ra/v1.0/cs01/soa-ra-v1.0-cs01.pdf> (accessed 02.03.2016)
- [11] The Open Group. “SOA Reference Architecture,” pp. 192, Nov. 2011. Available: <https://www2.opengroup.org/ogsys/jsp/publications/PublicationDetails.jsp?publicationid=12490>
- [12] OMG, “Service Oriented Architecture Modeling Language (SoaML®), Version 1.0.1,” May 2012. Available: <http://www.omg.org/spec/SoaML/1.0.1/>
- [13] E. Yu, “Models for Supporting the Redesign of Organizational Work,” in Proc. Conference on Organizational Computing Systems (COCS 1995), ACM press, pp. 226–236, 1995. Available: <https://doi.org/10.1145/224019.224044>
- [14] M.H. Danesh and E. Yu, “Modeling Enterprise Capabilities With i*: Reasoning on Alternatives,” in Proc. the Advanced Information Systems Engineering Workshops: CAiSE 2014 International Workshops, Thessaloniki, Greece, June 16–20, 2014, Springer LNBI 178, pp. 112–123, 2014. Available: https://doi.org/10.1007/978-3-319-07869-4_10
- [15] M.H. Danesh and E. Yu, “Analyzing IT Flexibility to Enable Dynamic Capabilities,” in International Conference on Advanced Information Systems Engineering, CAiSE Workshops 2015, Springer LNBI 215, vol. 3, pp. 53–65, 2015. Available: https://doi.org/10.1007/978-3-319-19243-7_5
- [16] S. Bērziša, G. Bravos, T. Gonzalez Cardona, U. Czubayko, S. España, J. Grabis, M. Henkel, L. Jokste, J. Kampars, H. Koç, J.-C. Kuhr, C. Llorca, P. Loucopoulos, R. Juanes Pascual, O. Pastor, K. Sandkuhl, H. Simic, J. Stirna and J. Zdravkovic, “Capability Driven Development: An Approach to Designing Digital Enterprises,” Business and Information Systems Engineering (BISE), vol. 57, no. 1, pp. 15–25, 2015. Available: <https://doi.org/10.1007/s12599-014-0362-0>
- [17] Oxforddictionaries.com, “Capability, Oxford Advanced Learner’s Dictionary”. Available: <http://www.oxforddictionaries.com/definition/learner/capability>
- [18] J.D. Collis, “Research Note: How Valuable are Organizational Capabilities?” Strategic Management Journal, John Wiley & Sons, vol. 15, pp. 143–152, 1994. Available: <https://doi.org/10.1002/smj.4250150910>
- [19] D.J. Teece, G. Pisano and A. Shuen, “Dynamic Capabilities and Strategic Management,” Strategic Management Journal, vol. 18, no. 7, pp. 509–533, 1997. Available: [https://doi.org/10.1002/\(sici\)1097-0266\(199708\)18:7<509::aid-smj882>3.0.co;2-z](https://doi.org/10.1002/(sici)1097-0266(199708)18:7<509::aid-smj882>3.0.co;2-z)
- [20] G.D. Bhatt and V. Grover, “Types of Information Technology Capabilities and Their Role in Competitive Advantage: An Empirical Study,” Journal of Management Information Systems, vol. 22, no. 2, pp. 253–277, 2005.
- [21] S. Bērziša, G. Bravos, T. Gonzalez, U. Czubayko, S. España, J. Grabis, M. Henkel, L. Jokste, J. Kampars, H. Koç, J.-C. Kuhr, C. Llorca, P. Loucopoulos, R. Juanes Pascual, K. Sandkuhl, H. Simic, J. Stirna and J. Zdravkovic, “Deliverable D1.4: Requirements Specification for CDD,” CaaS – Capability as a Service for Digital Enterprises, FP7 Proj. 611351, Riga Technical University, pp. 123, 2014. Available: <https://drive.google.com/file/d/0B8zbXOx8DWQqZHBYaWg3T3EzX1U/view>
- [22] K. Sandkuhl, J. Stirna, A. Persson and M. Wißotzki, “Enterprise Modeling: Tackling Business Challenges With the 4EM Method,” Springer-Verlag Berlin Heidelberg, 1st edition, 2014. Available: <https://doi.org/10.1007/978-3-662-43725-4>
- [23] G. Bravos, J. Grabis, M. Henkel, L. Jokste and J. Kampars, “Supporting Evolving Organizations: IS Development Methodology Goals,” 13th International Conference on Perspectives in Business Informatics Research, BIR 2014; Lund; Sweden, September 22–24, LNBI 104, pp. 158–171, 2014. Available: https://doi.org/10.1007/978-3-319-11370-8_12
- [24] J. Grabis, M. Henkel, L. Jokste, J. Kampars, H. Koç, K. Sandkuhl, D. Stamer, J. Stirna, F. Valverde and J. Zdravkovic, “Deliverable 5.3: The Final Version of Capability Driven Development Methodology,” FP7 Proj. 611351 CaaS – Capability as a Service in digital enterprises, Stockholm University, pp. 266, 2016. Available: <https://doi.org/10.13140/RG.2.2.35862.34889>

- [25] B.D. Rouhani, M.N. Mahrin, F. Nikpay and P. Nikfard, "A Comparison Enterprise Architecture Implementation Methodologies," 2013 International Conference on Informatics and Creative Multimedia, 4–6 Sept. 2013, pp. 1–6, 2013. Available: <https://doi.org/10.1109/iciem.2013.9>
- [26] J. Zachman, "The Concise Definition of The Zachman Framework by: John A. Zachman," The Official Concise Definition," 2009. Available: <https://www.zachman.com/about-the-zachman-framework>
- [27] Federal CIO Council, "Federal Enterprise Architecture (FEA) Framework v2," 2013. Available: https://obamawhitehouse.archives.gov/sites/default/files/omb/assets/egov_docs/fea_v2.pdf
- [28] R. Knaster and D. Leffingwell, "SAFe 4.0 Distilled: Applying the Scaled Agile Framework for Lean Software and Systems Engineering," 1st Edition, Addison-Wesley Professional, pp. 384, 2015.
- [29] E.B. Hyder, K.M. Heston and M.C. Paulk, "eSourcing Capability Model for Service Provider," Practice Details, ITSq Series, Carnegie Mellon, 2006.
- [30] M.C. Paulk, B. Curtis, M.B. Chrissis and C.V. Weber, "Capability Maturity Model SM for Software, Version 1.1.," Software Engineering Institute, Carnegie Mellon University, Pittsburgh, Pennsylvania 15213 Pittsburgh, Technical Report CMU/SEI-93-TR-024, ESC-TR-93-177, 1993.
- [31] W. Ulrich and M. Rosen, "The Business Capability Map: Building a Foundation for Business/IT Alignment," Cutter Consortium for Business and Enterprise Architecture. Available: <http://www.cutter.com/content-and-analysis/resource-centers/enterprise-architecture/sample-our-research/ea110504.html> (accessed 28.02.2016)
- [32] V. Mojtahed, U. Franke, M. Cohen, L. Luotsinen, H. Asadi and M. Eklöf, "Verktystöd för MBFU ur ett M&S-perspektiv," FOI Totalförsvarets forskningsinstitut, Linköping, Sweden, ISSN 1650-1942.
- [33] IBM, "Modeling With SoaML, the Service-Oriented Architecture Modeling Language, Part 1. Service Identification," IBM Developer Works, Jan. 2010. Available: <http://www.ibm.com/developerworks/rational/library/09/modelingwithsoaml-1/>
- [34] S. Brinkkemper, M. Saeki and F. Harmsen, "Meta-Modelling Based Assembly Techniques for Situational Method Engineering," Information Systems, vol. 24, no. 3, pp. 209–228, 1999. Available: [https://doi.org/10.1016/s0306-4379\(99\)00016-2](https://doi.org/10.1016/s0306-4379(99)00016-2)
- [35] C.L.B. Azevedo, M.E. Iacob, J.P.A. Almeida, M. van Sinderen, L.F. Pires and G. Guizzardi, "Modeling Resources and Capabilities in Enterprise Architecture: A Well-Founded Ontology-Based Proposal for ArchiMate," Journal of Information Systems, Elsevier. vol. 54, pp. 235–262, 2015. Available: <https://doi.org/10.1016/j.is.2015.04.008>