

# A Comparative Analysis of Concepts for Capability Design Used in Capability Driven Development and the NATO Architecture Framework

Janis Stirna<sup>(✉)</sup>

Department of Computer and Systems Sciences,  
Stockholm University, Postbox 7003, 164 07 Kista, Sweden  
js@dsv.su.se

**Abstract.** Recently an approach for information system design and delivery according to run-time context has been developed. It uses the concept of capability to express the organization's ability and capacity that enables it to achieve a business goal in a certain context, and hence is denoted Capability Driven Development (CDD). The concept of capability has also been used in other approaches because it facilitates business investment focus, it can be used as a baseline for business planning, and it directly leads to service specification and design. For example, several Enterprise Architecture frameworks have included capability as a key concept for analyzing organization's abilities to deliver desired functions. A notable contribution in this area is The NATO Architecture Framework (NAF), which aims at being a de facto standard for organizations operating in the areas of NATO. This paper analyses the possibilities of mapping the CDD concepts to NAF concepts that are relevant for capability design.

**Keywords:** Capability · Enterprise architecture · Enterprise modeling

## 1 Introduction

Linguistically, capability means *the ability or qualities necessary to do something* [1]. The notion of capability has been discussed throughout the past decades in application areas such as competence-based management, enterprise architecture management, developing firm's competitive advantage [2, 3], and, lately, Business-IT alignment [4]. Following the principles for specifying the organization's capacity and abilities to perform a business function, an approach called Capability Driven Development (CDD) has been developed [5].

CDD addresses the need for today's information system (IS) development frameworks and methodologies to support organizations acting in highly competitive and volatile environments, e.g. dealing with unexpected events, such as unpredicted increase of customer demands, legislation changes, new customer types, new alliances and competitors. The current business trends require companies to be able to operate continuously in dynamically changing business conditions [6, 7].

The CDD methodology is based on Enterprise Modeling (EM), context modeling, variability modeling, adjustment algorithms, and patterns for capturing best practices. It is also supported by the CDD environment that allows capability design and runtime monitoring and adjustment.

Capability is used in a wide variety of approaches and frameworks and while there are clearly identifiable similarities, there are also substantial differences in its use. If CDD is to succeed in being adopted by a broader community it needs to address standardization in the respective area of development. There are four basic options to consider, namely,

- (1) to propose a new standard,
- (2) to influence an existing standard, i.e. to make it change or to incorporate a new proposal,
- (3) to align with an existing standard or standards, or
- (4) do nothing related to standards, in the hope that the industry take-up will be fast and widespread enough to establish a de-facto standard in a “grass-roots” way.

Options 1 and 2 are unrealistic to carry out without significant and probably world-wide backing of key industrial players, especially in areas where existing standards are already developed. We consider the areas of IS development and Enterprise Architecture (EA) which are the primary targets of the CDD methodology being such. Hence, applying Option 3 of analyzing the existing landscape of standards and proposing alignments of CDD with the more prominent contributions in the field is the most appropriate path to choose. For a product such as the CDD methodology and environment, Option 4 is too risky because there are a significant number of standards existing in this area.

To this end, the objective of the paper is *to analyze how capability is addressed by one prominent EA framework, namely, The NATO Architecture Framework (NAF)*. The aim of this analysis is to show how CDD modeling components correspond to those of NAF in order to support application scenarios such as analyzing enterprise architectures documented according to NAF and then operationalizing them with CDD in order to monitor and adjust the runtime execution.

The remainder of the paper is organized as follows. Section 2 briefly presents the approaches and frameworks that use the concept of capability, including more details of CDD and NAF. The analysis of CDD and NAF concepts is provided in Sect. 3. Concluding remarks are presented in Sect. 4.

## 2 Background to Approaches Using Capability

The notion of capability has a growing presence in the current business and IT alignment and development frameworks. It is used by business-oriented frameworks, such as Business Architecture and Business Modeling as well as in the alignment-oriented frameworks for Enterprise Architecture (EA) and Enterprise Modeling (EM). The inclusion of the capability notion in the current development frameworks seems to have the following intensions: (a) for business planning, it is becoming recognized as a key component for describing strategies of what a core business does in order to deliver

value; (b) for IS development, it makes IS designs more understandable to business stakeholders by enabling them to use the capability notion to describe their requirements; and (c) for IS run-time and maintenance, it supports configurability of operations on a higher level of abstraction than services, process, and components.

The following areas of development approaches using the concept of capability can be identified:

- **OMG Business Architecture (BA) [8].** It is an enterprise blueprint aiming to provide a common understanding of an organization, as well as to align strategic objectives with tactical demands.
- **OMG Value Delivery Modeling Language (VDML) [9].** It defines a modeling language for analysis and design of the operations of an enterprise with a focus on the creation and exchange of value.
- **The Open Group Architecture Framework (TOGAF) [10]** describes architecture domains, concepts and methods for designing, using, and maintaining an enterprise architecture.
- **ArchiMate is an EA modeling language [11].** 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.
- **The Department of Defense Architecture Framework (DODAF) [12]** is an architecture framework for the US Department of Defense that provides visualization infrastructure for specific stakeholders concerns organized by various viewpoints.
- **The UK Ministry of Defence Architecture Framework (MODAF) [13]** is an architecture framework that defines a standardized way of conducting EA originally developed by the UK Ministry of Defence.
- **The NATO Architecture Framework (NAF) [14]** is an EA framework developed by NATO. It will be described in more detail in Sect. 2.2. 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. Considering at least the nominal intention of NAF being used by the large number of the NATO countries we have chosen NAF for the purpose of this analysis.
- **OASIS SOA provides an abstract, foundation reference architecture addressing the ecosystem viewpoint for building and interacting within the SOA paradigm [15].**
- **The Open Group SOA Reference Architecture defines a consumer and provider perspective with cross-cutting concerns describing architecture building blocks and principles that support the realizations of SOA [16].**
- **SOA Modeling Language (SoaML) by OMG defining a small set of extensions to UML to support SOA modeling [17];** it can be seen as an instantiation of a subset of The Open Group's architecture for representing SOA artifacts in UML.

A more detailed analysis of how the above frameworks address capability in terms of modeling perspective, definition, purpose, and methodology is available in [18].

## 2.1 Capability Driven Development Methodology

The CDD methodology consists of method components [5]. To structure the methodology, the components have been divided into upper-level method components and method extensions. Each upper-level component describes a certain application area and may also contain sub-components. The upper-level method components are currently the following:

- *Capability Design Process* guiding how to design, evaluate, and develop capabilities by using process models, goal models, and other types of models.
- *Enterprise Modeling* guiding the creation of enterprise models that are used as input for capability design.
- *Context Modeling* analyzing the capability context and its variations needed to deal with business process variations.
- *Reuse of capability design* guiding the elicitation and documentation of patterns for capability design.
- *Run-time Delivery Adjustment* adjusting capability at runtime.

The overall CDD process includes three cycles (1) capability design; (2) capability delivery; and (3) capability refinement/updating. The capability design cycle often starts with *Enterprise Modeling*, i.e. by a business request for a new capability - the request might be initiated by strategic business planning, changes in context, or discovery of new business opportunities requiring reconfiguration of existing or the creation of new goals, business processes or services, and other EM elements. This is followed with a formalized definition of requested capabilities and definition of the relevant contexts according, linking with relevant capability delivery patterns, as well as supporting IT applications all of which as can be seen as part of *capability design*.

In addition, several method extensions addressing specific business challenges to which the CDD methodology have been developed.

CDD defines capability as “*the ability and capacity that enable an enterprise to achieve a business goal in a certain context.*” The theoretical and methodological foundations for CDD is provided by the core capability meta-model (CMM) in Fig. 1, and in details presented in [5]. The CMM is developed on the basis of requirements from the industrial project partners, and related research on capabilities. In brief, the meta-model has three main sections:

- (a) *Enterprise model*, representing organizational designs with Goals, KPIs, Processes (with concretizations as Process Variants), and Resources. Key aspects are capability and goal dependency as well as capability and business process dependency showing intentional and operational aspects of each capability.
- (b) *Context*, represented with Context Set for which a Capability is designed and Context Situation at runtime that is monitored and according to which the deployed solutions should be adjusted. Context Indicators are used for measuring the context properties (Measuring Property); and
- (c) *Patterns*, for delivering Capability by reusable solutions for reaching Goals under different Context Situations. Each pattern describes how a certain Capability is to be delivered within a certain Context Situation and what Processes Variants and Resources are needed to support a Context Set.

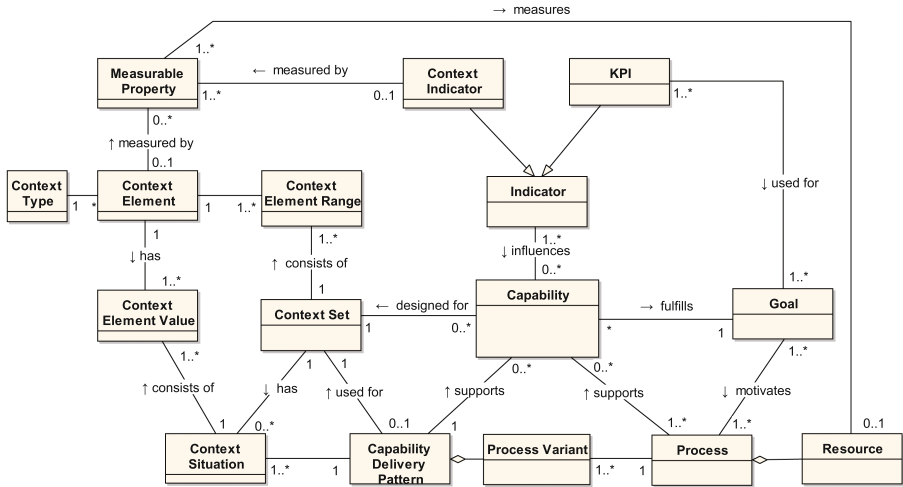


Fig. 1. A core meta-model for supporting Capability Driven Development [19].

Figure 1 is a simplified version of the CMM showing only the key components of CDD and omitting, for instance, constructs for representing goal decomposition relationships and process variants. The complete version including definitions of the components is available in [5, 19].

## 2.2 The NATO Architecture Framework

NAF is organized into a number of views such, All Views, Capability View, Operational View, System View, Service View, Technical View, and Programme View. Capability view, further specified in models addressing detailed aspects of capability development, namely: Capability taxonomy (C1), Enterprise Vision (C2), Capability dependencies (C3), Standard Processes (C4), Effects (C5), Performance Parameters (C7), Planning Assumptions (C8), and Capability Roadmap (Cr).

NAF defines capability as “the ability of one or more resources to deliver a specified type of effect or a specified course of action” [14]. Examples of capabilities according to NAF are “Tank production, 20 tanks per year”, “Tank production, 20–40 tanks per year”, “Light Armor Vehicle Recovery”, and “Heavy Armor Vehicle Recovery”. Details of the capability definition and associations is given in the NAF meta-model [14], see Figs. 2 and 3. The key associations of capability are as follows:

- Capabilities may be specialized into more specific capabilities, composed of several capabilities, as well as dependent on other capabilities.
- Capability when applied is associated with measurable categories
- Capability elaborated into Capability configuration package, which is used to configure resources for capability implementation.

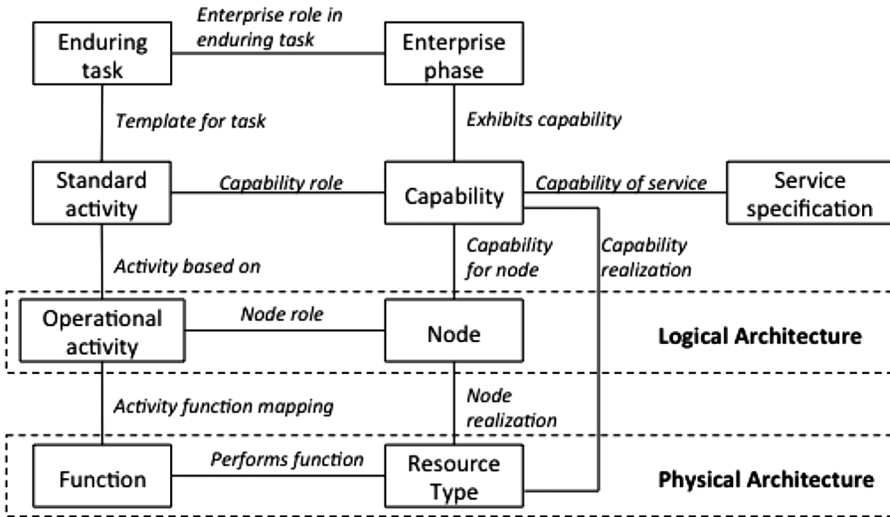


Fig. 2. A simplified overview of the NAF meta-model, adapted from [14].

- 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 the task.

A more detailed meta-model of the Enterprise Vision view (C2) is shown in Fig. 3 on the basis of which the analysis of CDD and NAF was performed. The purpose of a C2 view is to provide a strategic context for the capabilities described in the architecture and to specify the scope for the architecture in terms of vision, goals, enduring tasks and capabilities.

NAF is interoperable with MODAF because it is based on earlier versions of MODAF. The interactive website of the NAF meta-model also has the ability to present NAF according to MODAF views. Hence this analysis is relevant even to those countries that are not part of NATO and use, as in the case of Swedish Armed Forces, MODAF instead.

### 3 Analysis of CDD and NAF Concepts

NAF addresses capability in the Capability Viewpoints (C1-C8). This section will analyze the key concepts of these viewpoints as defined in the NAF meta-model (see Figs. 2 and 3) with respect to the CDD meta-model (Fig. 1).

We will analyze the concepts of the NAF meta-model that are relevant to the five key areas addressed by CDD method components, namely, Capability Design (including Enterprise Modeling), Context Modeling, Business Process and Variability Modeling, Reuse and Patterns, as well as Adjustment Algorithms. These are mostly documented in NAF viewpoints C1 to C8 and Cr. Components from these viewpoints have been analyzed only if they have direct influence on the concept of capability, i.e. NAF

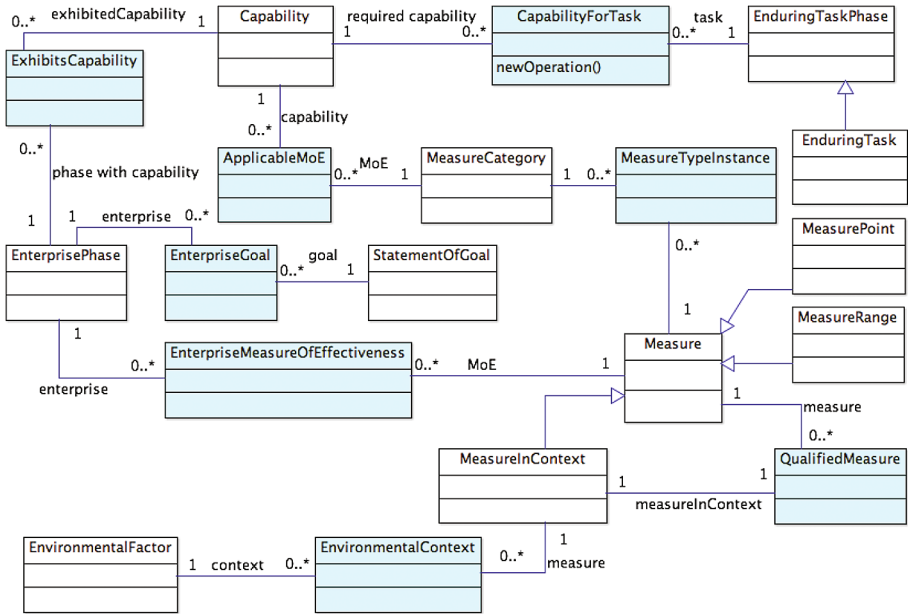


Fig. 3. NAF meta-model for view C2 Enterprise Vision, adapted from [14].

viewpoints addressing, for instance, services and other aspects of the architecture have been excluded from this paper.

Figure 3 shows a part of the meta-model of view C2 Enterprise Vision that have relations with capability and addresses similar components to the CMM. It uses the UML Class diagram notation; classes in color denote objectified relationships.

Table 1 shows the proposal for alignment of NAF concepts and CDD concepts. Considering the high complexity of the NAF meta-model we also indicate in which view of the meta-model the concept is chiefly used. The CDD meta-model is not structured in views although each of its method components focuses on a specific part of modeling and hence can be considered as dynamic views.

Table 1 shows constructs of both approaches that have similar purposes and are in principle interchangeable in the sense that they serve similar purposes. In several cases of mapping the NAF concepts to CDD the mapping would depend on the intensions of the modeling. E.g. in the case of Enduring task, since it is defined as “a specification of what the enterprise does”, it would be our primary choice of modeling it with a CDD business process on a high level of decomposition. This would support clear visibility of how the process and process variants are used to deliver a capability. Another option, however, would be to model it with an operational goal, which would make the CDD model more understandable from a strategic perspective.

The constructs of NAF cover three of the key aspects of CDD, namely, Capability Design (including Enterprise Modeling), Context Modeling, and Business Process and Variability Modeling. The forth aspect of Reuse and Patterns is not explicitly addressed by NAF, but it can be covered by developing service specifications (NAF viewpoints

**Table 1.** A proposal for alignment of NAF and CDD concepts

NAF Concept	NAF view-point	Relevant CDD concept	Comments
Capability	C1	<i>Capability</i>	The main difference is that NAF definition of capability does not include the context dependence
Capability specialization relationship	C1	<i>Capability composition relationship</i>	NAF uses specialization to specify more generic capabilities into more specific capabilities. According to CDD all capabilities should be operational (not abstract) hence, composition of capability into more atomic sub-capabilities is more appropriate
Capability composition relationship	C3	<i>Capability composition relationship</i>	Both NAF and CDD have a relationship for specifying that one capability can be a part of another
Capability dependency relationship	C3	<i>Capability collaboration</i>	In NAF capability dependency relationship is used for analyzing the dependencies between capabilities and between capability clusters. CDD has a method extension for modeling capability collaboration, for a similar purpose
Measure Category	C1, C2	<i>Context Indicator KPI</i>	Measure Category is used for specifying Measures of Effectiveness relevant to Capabilities. CDD monitors capability by KPIs and Context Indicators both of which are based on Measurable properties
Capability relationship related to Measure Category	C2, C7	<i>Capability relationship influences Indicator</i>	According to NAF Capabilities are measured according to Measure Categories which are specified as Measures of Effectiveness (MoE) and are manifested by Measures when executed. This is similar to CDD measurement according to Indicators, which can be of two kinds – Context Indicator and KPI

(continued)



**Table 1.** (continued)

NAF Concept	NAF view-point	Relevant CDD concept	Comments
Statement of goal	C2	<i>Goal</i>	NAF and CDD goals are similar in their definition and relationships for goal dependencies and refinement
Goal has sub-goal relationship	C2	Goal refinement relationships ( <i>motivates, hinders, AND, OR</i> )	CDD is based on 4EM and hence has a richer set of relationships for goal modeling. This can be explained by the fact that NAF is not intended for extensive goal modeling
Vision statement	C2	<i>Goal</i>	NAF's Vision Statement is a short paragraph outlining the vision for a given phase of an enterprise; it is a part of Enterprise Phase. In CDD terms, this is modeled by goals and the elaboration of a goal hierarchy
Enterprise Phase	C2		In NAF Enterprise Phase denotes an UndertakingState that is a current or future state of a WholeLifeEnterprise, e.g. as-is or to-be. CDD does not have a specific construct for modeling phases of organization transformation. This is usually achieved by structuring the model into views and sub-models
Measure	C2	<i>Measureable property, Context Element</i>	In NAF Measure can be specialized into Measure In Context which in turn can be linked to Environmental Factor, and into Measure Range. These concepts are similar to the ones provided by the Context Modeling method component of CDD, specifically Measureable Property, Context Element, and Context Element Range
Measure in Context	C2	<i>Context Element, Measurable Property</i>	NAF has Measure In Context as a measure in a specific Environmental Factor such as Terrain Type, Weather etc. This can be modeled with the CDD

(continued)

**Table 1.** (continued)

NAF Concept	NAF view-point	Relevant CDD concept	Comments
			concepts Context Element and Measurable Property
Capability relation to Enterprise Phase	C2		In CDD, phases of the change management process are realized by model structure, hence such relationship is not present on a conceptual level
Enduring task	C2	<i>Process</i>	In NAF Enduring task is seen as an undertaking recognized by an enterprise as being essential to achieving its goals - i.e. a strategic specification of what the enterprise does. This concept is similar to CDD business Process at a higher level of decomposition. I.e. capability is delivered by a business process, but the actual context variations are achieved by process variants
Capability relation to Enduring Task	C2	<i>Capability requires Process</i>	In NAF Enduring task is a sub-class of Enduring Task Phase, and the purpose of the relationship is to specify what Capability is required in order for an Enterprise to conduct a phase of an Enduring Task. In CDD this is realized by the relationship Capability requires Process
Standard Activity	C4	<i>Process</i>	NAF defined Standard Activity as a ProcessType that is a standard procedure (e.g. doctrinal tasks). Standard activity can consist of other Standard Activities and is based on Operational Activity. This is modeled as Process in CDD
Capability relationship has role in Standard Activity	C4	<i>Capability requires Process</i>	In NAF this relationship is used to specify that a Capability participate in a StandardActivity. In CDD the same can be achieved by tracing all relationships between Capability and Processes and Process Variants

S1 to S8). The fifth aspect of CDD, namely, Adjustment Algorithms, is not explicitly addressed by NAF and hence additional workarounds might need to be used for instance in the Logical viewpoints.

Concerning the way of working, CDD offers well elaborated method guidance and extensive supporting material. In contrast, currently the method guidance for NAF is based on the Architecture Development Method (ADM) of TOGAF, but it is still a work in progress.

In terms of tool support CDD offers an integrated environment for capability design and runtime monitoring and adjustment while the tools that support NAF primarily focus on architecture design and documentation. The current status of the CDD environment allows adjustment and configuration of existing systems, such as ERP systems. To support cases when a new IS needs to be developed to realize capability delivery, an ongoing work on supporting capability designs with Model Driven Development is reported in [20].

## 4 Concluding Remarks

In this paper we have summarized how the concept of capability is used in CDD and NAF for the purpose of aligning CDD with an established EA framework. While there are differences in capability definitions, meta-models, and the way it should be modeled, both use capability to bind the intentional part of the organizational design with the operational part that also encompasses variability/alternatives.

We can conclude that NAF has a more generic scope; hence it includes more constructs for strategic planning and specification of IT architecture. CDD mostly focuses on operational capability design and execution of the adjustments. Hence CDD is more streamlined when it comes to modeling enterprise designs in terms of goals, processes, resources, and best practices (patterns). CDD also supports monitoring capability performance in terms of context elements and KPI as well as specification of adjustment algorithms that are automatically deployed in the Capability Navigation Application. While NAF defines measures of effectiveness, development of monitoring and adjustment applications can be seen as aspects of Model Driven Development, which beyond the main purpose of NAF and hence it does not explicitly support them.

The overall approach taken in the project that developed the CDD methodology has been to focus on the elaboration of proposals for CDD alignment with significant standards that are used in practice which in turn support broader adoption of CDD. The analysis performed in this paper supports the application scenario of elaborating capability designs with CDD from existing organizational designs expressed according to NAF. Such a way of working would contribute to implementing the aspects of runtime monitoring and adjustment of information systems according to context changes for which NAF currently does not offer explicit support.

## References

1. Oxforddictionaries.com. Capability, Oxford Advanced Learner's Dictionary. <http://www.oxforddictionaries.com/definition/learner/capability>
2. Collis, J.D.: How valuable are organizational capabilities? *Strateg. Manag. J.* **15**, 143–152 (1994). Wiley
3. Teece, D.J., Pisano, G., Shuen, A.: Dynamic capabilities and strategic management. *Strateg. Manag. J.* **18**(7), 509–533 (1997)
4. Bhatt, G.D., Grover, V.: Types of information technology capabilities and their role in competitive advantage: an empirical study. *J. Manag. Inf. Syst.* **22**(2), 253–277 (2005)
5. Grabis, J., Henkel, M., Kampars, J., Koç, H., Sandkuhl, K., Stamer, D., Stirna, J., Valverde, F., Zdravkovic, J.: D5.3 The final version of Capability driven development methodology, FP7 proj. 611351 CaaS – Capability as a Service in digital enterprises, doi:[10.13140/RG.2.2.35862.34889](https://doi.org/10.13140/RG.2.2.35862.34889)
6. Lucke, C., Krell, S., Lechner, U.: Critical issues in enterprise architecting - a literature review. In: AMCIS 2010 (2010). <http://aisel.aisnet.org/amcis2010/305>
7. Stirna, J., Zdravkovic, J.: Interview with Sladjan Maras on challenges and needs in enterprise modeling. *BISE J.* **57**(1), 79–81 (2015)
8. OMG Business Architecture Special Interest Group&Guild. “What Is Business Architecture?” <http://bawg.omg.org/>, <http://www.businessarchitectureguild.org/>
9. OMG: Value Delivery Metamodel, v. 1.0 (2013). <http://www.omg.org/spec/VDML/1.0/>
10. The Open Group, TOGAF 9.1, a Standard. The Open Group. <http://pubs.opengroup.org/architecture/togaf9-doc/arch/index.html>
11. The Open Group. ArchiMate 2.1 Specification. The Open Group, December 2013. <http://www.opengroup.org/archimate/downloads.htm>
12. Department of Defense DoD Architecture Framework Version 2.0, 28 May 2009. [http://dodcio.defense.gov/Portals/0/Documents/DODAF/DoDAF\\_v2-02\\_web.pdf](http://dodcio.defense.gov/Portals/0/Documents/DODAF/DoDAF_v2-02_web.pdf)
13. UK Ministry of Defence, MOD Architecture Framework (2012). <https://www.gov.uk/guidance/mod-architecture-framework>
14. UK Ministry of Defence, NATO Architecture Framework v4.0 Documentation (2013). <http://nafdocs.org/modem>
15. OASIS. SOA Reference Architecture 1.0. <http://docs.oasis-open.org/soa-rm/soa-ra/v1.0/cs01/soa-ra-v1.0-cs01.pdf>. Accessed 02 Mar 2016
16. The Open Group. SOA Reference Architecture, November 2011. <https://www2.opengroup.org/ogsys/jsp/publications/PublicationDetails.jsp?publicationid=12490>
17. OMG. Service oriented architecture Modeling Language (SoaML®), Version 1.0.1, May 2012. <http://www.omg.org/spec/SoaML/1.0.1/>
18. Zdravkovic, J., Stirna, J., Grabis, J.: A comparative analysis of using the capability notion for congruent business-and information systems engineering. *Complex Syst. Inf. Model. Q., CSIMQ*, **10**, 1–20 (2017)
19. Bērziša, S., Bravos, G., Gonzalez Cardona, T., Czubyko, U., España, S., Grabis, J., Henkel, M., Jokste, L., Kampars, J., Koç, H., Kuhr, J.-C., Llorca, C., Loucopoulos, P., Juanes Pascual, R., Pastor, O., Sandkuhl, K., Simic, H., Stirna, J., Zdravkovic, J.: Capability driven development: an approach to designing digital enterprises. *Bus. Inf. Syst. Eng. (BISE)*, **57**(1) (2015). doi:[10.1007/s12599-014-0362-0](https://doi.org/10.1007/s12599-014-0362-0)
20. Henkel, M., Stratigaki, C., Stirna, J., Loucopoulos, P., Zorgios, Y., Migiakis, A.: Combining tools to design and develop software support for capabilities. *Complex Syst. Inf. Model. Q., CSIMQ*, (10), 38–52 (2017). <https://doi.org/10.7250/csimq.2017-10.03>