

Causal Modelling in Enterprise Architecture Frameworks

Saulius GUDAS

*Institute of Data Science and Digital Technologies, Vilnius University, Akademijos str. 4,
LT-08663 Vilnius, Lithuania
e-mail: saulius.gudas@mif.vu.lt*

Received: May 2020; accepted: March 2021

Abstract. The paper deals with the causality perspective of the Enterprise Architecture (EA) frameworks. The analysis showed that there is a gap between the capabilities of EA frameworks and the behavioural characteristics of the real world domain (enterprise management activities). The contribution of research is bridging the gap between enterprise domain knowledge and EA framework content by the integration of meta-models as part of EA structures. Meta-models that cover not only simple process flows, but also business behaviour, i.e. causality of the domain, have been developed. Meta-models enable to create a layer of knowledge in the EA framework, which ensures smart EA development, allows validation of developer decisions. Two levels of the enterprise causal modelling were obtained. The first level uses the Management Transaction (MT) framework. At the second level, deep knowledge was revealed using a framework called the Elementary Management Cycle (EMC). These two causal frameworks were applied here to justify the causal meta-models of the EA. The new concepts *Collapsed Capability*, *Capability Type* and *Capability Role* which meaningfully complement MODAF with causal knowledge are introduced. Strategic Viewpoint (StV) modelling using causal meta-models is described in detail and illustrated in the case study. The example provided shows a principled way that causal knowledge supports the verification and validation of EA solutions. The presented method provides an opportunity to move the EA development to smart platforms.

Key words: Enterprise Architecture framework, causality, causal modelling, Management Transaction, capability, MODAF, causal meta-model.

1. Introduction

Causality and causal knowledge are key concepts in science that underpin the development of smart, automatic, autonomous and intelligent systems. Causality is an important concept in modern science (Bunge, 2011); it helps to reveal the domain properties hidden to the outside observer. Causal knowledge is a type of knowledge, next to declarative, procedural, and relational knowledge. Causal knowledge is a “description of causal links among a set of factors . . . which provides a means for organizations . . . how best to achieve some goal” (Zack, 1999).

Causality as a theoretical concept is discussed in Schurz and Gebharder (2016). According to Glymour (2004), Schurz and Gebharder (2016) causality should be understood

as a theoretical concept (in analogy to the concept of force in Newtonian physics). A general theory of causation was developed by Pearl (2000, 2009), which underlies the theory of causal nets (TCN) developed in Spirtes *et al.* (2000), Pearl (2009). Two notions of causality can be distinguished – type causality (so-called general causality) and actual causality (called specific causality) (Halpern, 2015). Actual causality focuses on particular events, while type causality is looking for common regularity (law). The understanding of causality in system modelling can be quite different according to the nature of the knowledge used.

The awareness of the theory of specific domain causality is the prerequisite for discovering deep knowledge (i.e. regularities, laws) in a given domain. Causation methods are common in statistics, econometrics, cybernetics, computer science, data science and other complexity sciences to study cause-effect relationships and construct causal models in order to predict and control the possible dynamics of the systems. Causal models are tailored to a specific type of domain, describing regularities specific to that area of reality, e.g. such as physical systems (work centres, robots, autonomous devices), enterprises (production and business companies, education systems, etc.), biological systems, economic systems, ecological systems, etc.).

Excellent results have already been achieved in the development of the cyber-physical systems (CPS) such as smart systems, autonomic systems and other types of CPS. CPS engineering uses the intrinsic properties of a domain (i.e. the Physical System) because there is a long-established good theoretical foundation – control theory. In a physical system, causality is a dependency between causes (impacts, events, faults, etc.) and changes of a system (state, transition, parameter values, etc.). In other words, CPS engineering methods are based on the causal knowledge (scientific law, scientific explanation) of the subject domain (Bunge, 2011). It makes sense to look for the causal knowledge (scientific law, regularities) in other types of real world domains. Causality in risk management is to be considered as the direct relation of the event to a risk *situation* (influence relation) (Sienou *et al.*, 2008).

Our research area is business modelling and enterprise modelling. One of the definitions of causality in this area is as follows: the dependence of enterprise goals on components of the enterprise such as processes, material flow, information flow, services, systems, and so on (Lagerström *et al.*, 2009).

We are dealing with complex systems (organizational systems, enterprises or cyber-social systems) summarized by the term systems of systems in the methodology of EA frameworks. Such a subject domain is referred to as “enterprise” in application software engineering and is related to a wide range of industries, e.g. manufacturing, military, and healthcare, energy, communication enterprises, and others. A captured domain causality is specified as the internal domain model, e.g. the cause–consequence rules, equations, ontology, meta-model or other structures of causal knowledge (Grundspenkis, 1998).

Two levels of enterprise causal knowledge were introduced for software engineering needs in Gudas (2012). The first level is the presentation of the discovered causation using the Management Transaction (MT) framework. At the second level, a deep knowledge structure of MT is revealed in a more detailed framework called the Elementary Management Cycle (EMC). Causal modelling is suitable for discovering causal dependencies

in various real-world domains. These are not just organizational systems (i.e. enterprises, cyber-enterprises or cyber-social systems), but also biological systems (organisms), ecological systems, the content of education systems and other complex systems.

The aim of the study is to bridge the gap between causality inherent in the definite real world domain and EA framework structures.

The EA frameworks are currently based on expert knowledge and experience, these are generalized structures derived from the evolution of the EAF. The analysis showed that there is a gap between the capabilities of EA frameworks and the behavioural characteristics of the real world domain (enterprise management activities).

The aim is to improve the existing EA development methods and systems, use casual knowledge of the activity domain in validating the decisions of EA developers, developing intelligent EA development systems. The paper aims to improve the conceptual structure of the existing EA frameworks (MODAF, ArchiMate, etc.) using causal domain knowledge.

The influence of the newly created EAF component type – meta-models – on the EA development process is investigated. This study is linked to previously published work on the structure of the expanded MDA/MDD process (Gudas and Valatavičius, 2020). An extended structure of MDA is described here, where above the CIM layer is the modelling layer of the reality domain, called the “domain knowledge model”.

The relevance of EA frameworks for more accurate (deeper) modelling of business processes is emphasized (Schekkerman, 2004) – it is necessary to “use business behaviour instead of business processes as part of the EA framework”.

Causal modelling seeks to reveal the domain regularity and is consistent with the internal modelling paradigm. If an external modelling paradigm is applied, then the modelling is based on external observation, obtaining empirical information that does not reveal essential (deep) causal dependencies. The external modelling relies on the naming of “specific events” and its input-output analysis, and the true causality is not determined in this way.

From the perspective of internal modelling, an enterprise (organizational system) is a complex system with a self-managing property, the essence of which is the feedback (control) loop – the circular causality of elements (activities, processes or components).

Circular causality between elements is considered as a transaction and is formally defined as a wheel graph in Gudas and Valatavičius (2017, 2020). The transaction is an essential concept in computer science, database management systems, business modelling notation BPMN 2.0, transactional workflows (Medina-Mora *et al.*, 1992), enterprise management modelling using the transaction concept (Dietz, 2006; Gudas, 2012).

Circular causation is an essential feature of these conceptual enterprise models: Action Workflow model (Rusinkiewicz and Sheth, 1994; Medina-Mora *et al.*, 1992), Deming’s PDCA cycle of business management (Deming, 1993), ITIL Framework (Persse, 2012), or the autonomic computing component (Kephart and Chess, 2003).

Therefore, in our view, a transaction is defined as a conceptual model of circular causality – a description of an essential feature of an enterprise as a complex system. Thus, a transaction is a key concept that allows you to discover the deep characteristics (causality) of an enterprise domain.

The rationale for incorporating the causality paradigm into EA modelling language is the need for adequate modelling capability – to establish a circular causality in the enterprise architecture. As discussed above, the circular causality is an essential feature required to properly identify the self-managed enterprise activities and represent it as management transactions.

The problem (research question) is bridging the gap between domain causal knowledge and the content of the EA system by integrating meta-models as part of EA structures. Second, the necessary properties of meta-models need to be ensured – they must cover not only simple process flows, but also capture the causality of the company’s activities. Such a study includes the application of the extended MDA scheme developed (Gudas and Valatavičius, 2020). Meta-models are expected to create a layer of knowledge in the EA development repository, which ensures smart EA development, allows validation of developer decisions.

The basic concepts of causal enterprise modelling are described in more detail (Gudas, 2012, 2016): *Management Functional Dependency (MFD)*, *Management Transaction (MT)*, and *Elementary Management Cycle (EMC)*.

An analysis of the known EA frameworks and business process modelling notations was focused on the key concepts that determine the possibilities of causal knowledge representation. Known EA structures were examined, including MODAF, ArchiMate, and UAF (Morkevičius *et al.*, 2017; MODAF, 2013; ArchiMate, 2017), as well as key concepts of other modelling approaches – OMG standards BPMN, BPDM, BMM, OSM, goal-driven methods KAOS (Dardenne *et al.*, 1993), GBRAM (Anton, 1996), the NFR framework (Mylopoulos *et al.*, 1992). A comparison of the EA frameworks and other modelling standards allows us to identify the MODAF framework as the most comprehensive one. However, it also contains “white spots” as there are no suitable MODAF products to specify several aspects of enterprise management and validate the resulting models. One of the objectives is to extend MODAF using causal knowledge, complementing the strategy modelling and operational views. Upgrading the core EA systems with causal models (using the MODAF example) reveals the structure of intelligent EA development software. This would streamline the EA development process and increase the quality of the software development.

Several graphical notations were used in this work: UPDM notation to present EA constructs (UPDM, 2017), MODAF meta-modelling tagging (MODAF, 2013), DFD notation for conceptual models.

The remainder of the paper structured as follows: Section 2 explains paradigms of system modelling, the concepts of causal modelling, provides an overview of the enterprise architecture frameworks from a causal modelling perspective. Section 3 introduces the internal model of Porter’s value chain and the concepts of causal knowledge: Management Transaction (MT) and Elementary Management Cycle (EMC). The detailed structure of MT and the specific version of EMC for the enterprise management modelling are set out in Section 4. The causal constructs of enterprise architecture modelling are defined in Section 5. The assumptions for the development of causal EA are formulated in Section 6. Section 6 also provides description of the causality-based MDA/MDD process,

the architecture of intelligent EA tools, a causal EA development scheme aligned with the MODAF framework, the EA development stages on the causal CIM* layer of MDA, and types of validation based on causal knowledge. Additions to the MODAF Strategic viewpoint (StV) are the causal StV meta-models are presented in Section 7. Section 8 provides an example of causal Strategic Viewpoint StV development, includes rules for (vertical) model's transformations and (horizontal) validation processes. The conclusions summarize the essence of causal modelling for EA upgrading and the benefits of causal enterprise architecture development.

2. Related Works

2.1. Paradigms of System Modelling

Different system modelling paradigms are known, each with appropriate analysis and modelling techniques (Gudas and Valatavičius, 2020):

1. The paradigm of external modelling is related to the black box principle; in this paradigm, the analysis of the subject domain is based on external observation of inputs and outputs. The resulting model is based only on external observations, because a priori (scientific) knowledge about the regularities of the domain is not known or used.
2. The internal modelling paradigm is related to the white box principle, where the inner components and interactions are available for analysis. The internal modelling implies that the model is constructed using a priori (scientific) knowledge that completely describes the domain causality. In this paradigm, the subject domain analyst seeks to reveal and explore causal knowledge. If a priori (scientific) knowledge is incomplete, it only partially describes the causal relationship of the domain, that is, gray box modelling. We assign it here as well.

The level of awareness of the subject domain (regularities of internal interactions) is increasing when moving from black-box models towards grey-box and, finally, to white-box models. Illustrative examples of the concepts:

- External modelling (black box modelling – *external observation* based modelling): “*An airplane can fly because it has wings like a bird*”;
- Internal modelling (white box modelling – *knowledge gained through internal observation of causal dependencies*): “*the bird and the airplane fly because their wing configuration is appropriate and the law of aerodynamics is in effect*”.

Next, we describe each of these paradigms in more detail.

The external modelling paradigm. Externally observed processes (events or objects), relationships, inputs, and outputs are the elements that make up a model. The causes of relationships in such a model are not explicable, because there is no theory-related content for causal dependencies in the domain. More sophisticated external modelling methods

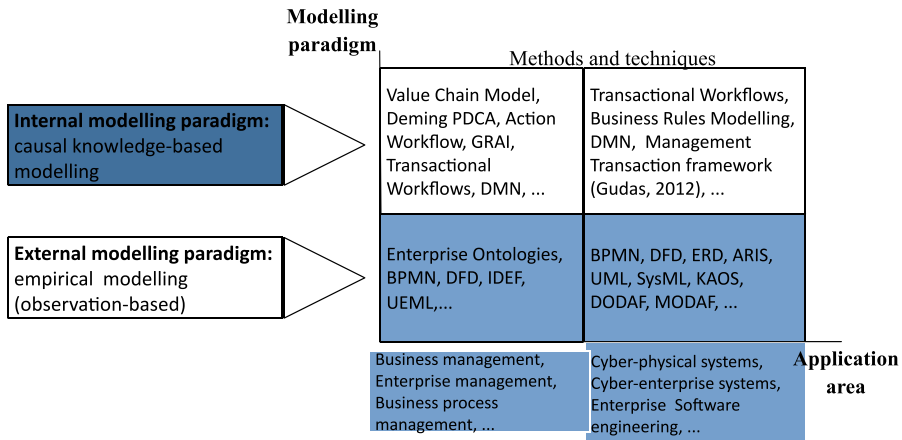


Fig. 1. A quadrant of domain modelling paradigms and methods.

use generalized domain models (meta-models, ontologies or patterns). However, these generalized frameworks are also based on external observation and have no theory of causality in the subject area.

Most of the business process modelling languages (Fig. 1) are attributed to the external modelling paradigm (Gudas and Valatavicius, 2017): BPMN, Data Flow Diagrams (DFD), IDEF, UML, SysML, Event-Process Chain (EPC) based ARIS method. All of them are designed to describe the results of the external observation, but not internal causation dependencies. This also applies to frameworks like UEML (Unified Enterprise Modelling Language), also enterprise architecture frameworks DoDAF, MODAF, UAF (Morkevičius and Gudas, 2011). Therefore, they rarely contain modelling concepts that help uncover a domain causality and understand causal dependencies. We notice that domain causation is more complex and with deeper knowledge than cause-effect interactions of activities, processes, functions, material and/or information flows perceived by external observation.

The qualitative difference between the concepts of external (empirical) modelling and internal (causal) modelling is expressed by the following examples:

- External modelling paradigm [black box modelling, observation-based modelling, empirical modelling]: an airplane can fly because the bird also has wings and therefore flies; the robot may step on and not overturn as a computer controls it;
- Internal modelling paradigm [white box modelling, cause-effect analysis, cause-effect sequence analysis, causal modelling]: a bird and an airplane fly only because their wing configuration is appropriate and the law of aerodynamics applies; the robot can move and not overturn because it is controlled by a software that has programmed the laws of control theory and physics.

Thus, causality is expressed as causal knowledge in the form of regularity, a consistent model (meta-model), physical law that is valid in a particular domain of reality.

Forster's remarkable note on circularity (circular causality) in complex systems is of particular importance today: "Should one name one central concept, a first principle, of cybernetics, it would be circularity" (Von Foerster *et al.*, 1953).

The circular causality can be exposed, for example, by using transactional workflows – a combination of workflow patterns and transaction models (Grefen, 2002; Injun *et al.*, 2002). Transactional workflow refers to a model in which a sequence of interactions goes from one workflow task (step) to another (or from one subsystem to another) and back to the first one (Rusinkiewicz and Sheth, 1994). A topology of the generalized transaction is compared to a wheel graph (Gudas and Valatavičius, 2020).

The internal modelling paradigm. Cybernetics and the emergence of complexity sciences have produced general descriptions that reveal the causation in complex systems such as social systems, biological systems, economical systems, and others.

We present concepts from other engineering and science disciplines that describe inherent causal dependencies:

- A *closed-loop control*, *self-regulation*, and *adaptation* are key concepts of system theory, control theory, and are the terms that cybernetics and complex system theory deals with;
- In biological systems, the term *homeostasis* denotes a self-regulating process by which biological systems tend to maintain its parameters that are required for survival within a normal range of values;
- In ecology research, the term *vicious circle* refers to a complex chain of events, which reinforce themselves through a feedback loop;
- In economics *sustainable development* deals with mutual dependencies (self-regulating processes) of four interconnected domains: ecology, economics, politics, and culture;
- The Rummler-Brache methodology of *managing the organization (enterprise) as an adaptive system* reveals a hierarchy of *management causal dependencies* – feedback loops (circularity) on the organizational level, the process level, and the job/performer level.

A circular causality of the management activities is uncovered in several frameworks: PDCA quality management cycle (Deming, 1993), Rummler-Brache enterprise management model (Rummler *et al.*, 2010), Value Chain Model (VCM) (Porter, 1985), business risk management standards (ISO:31000, 2009; OCEG “Red Book” 2.0, 2009, etc.), the enterprise transaction framework in DEMO (Dietz, 2006), Action Workflow (Medina-Mora *et al.*, 1992).

The listed and some other business management frameworks (e.g. Deming’s PDCA cycle) include feedback loop (circularity) as an essential construct (see Fig. 2a and 2b). In summary, they can be said to have been formally referred to as the management transaction (Fig. 2c) (Gudas and Valatavičius, 2020). All these business management frameworks consider enterprises as goal-driven systems. Typically, they include the *Goal* concept, which has an impact on other internal components. The self-management capability of enterprise is determined by the control feedback loop between physical processes and managerial activities (e.g. data or signal processing, decision-making, or computations). It is also important that the TOGAF framework (TOGAF, 2018) and ITIL framework (Persse, 2012) have similar topologies as the management transaction in Fig. 2c.

Therefore, in our view, a transaction is defined as a conceptual model of circular causality – a description of an essential feature of an enterprise as a complex system. Thus,

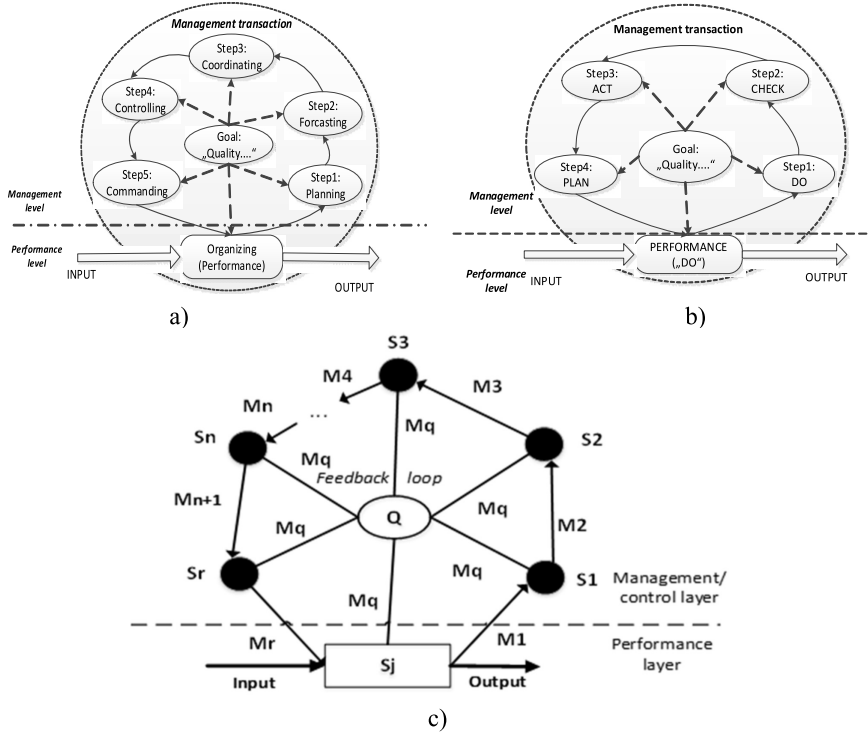


Fig. 2. Internal enterprise models (a business management viewpoint). a) Enterprise management cycle according to Fayol, b) Deming’s PDCA cycle, c) a represented Management Transaction is a kind of wheel graph.

a transaction is a key concept that allows you to discover the deep characteristics (causality) of an enterprise domain.

The rationale for incorporating the causality paradigm into modelling language is the need for adequate modelling capability – to establish a circular causality in the enterprise architecture. As discussed above, the circular causality is an essential feature required to properly identify the self-managed enterprise activities and represent it as management transactions.

2.2. Overview of Enterprise Architecture Frameworks

This section briefly discusses the architecture of the more advanced EA frameworks ArchiMate, DoDAF (Emery and Hilliard, 2009; DoD, 2007), MODAF, and UAF (Matthew *et al.*, 2013, 2016; Schekkerman, 2004). The analysis of EA frameworks was performed using the methodology developed by Kosanke (1997), which is based on the comparison of key concepts defined in the frameworks in question.

An open and independent performance architecture modelling language ArchiMate is a relatively small EA framework that is under development (ArchiMate, 2017). New expanded version ArchiMate 3.0 offers a generalized language for describing enterprise at

a strategic level (new layer), physical world of materials and equipment (new layer), business process structure, operations, organizational structure, information flows, IT systems and technical infrastructure (ArchiMate, 2017). This work explores the business process layer. The meta-model of this layer was created and the key concepts were extracted: *Goal, Capability, Resource, Outcome, Structure, Actor, Role, Interface, Collaboration, Behaviour, Service, Process, Function, Interaction, Event, Information, Object, Representation, Product, Meaning, Value* (Gelžinienė and Gudas, 2015).

Recognized enterprise architecture frameworks MODAF and DoDAF are very broad structures that you need to adapt to your context (Schekkerman, 2004). Because MODAF is a newer and more refined system, we pay more attention to it. MODAF defines a standard way of capturing business strategy, identifies associated capabilities and processes, and provides the basis for building the enterprise architecture needed to deliver the strategic vision (MODAF, 2013). This framework consists of seven viewpoints (Strategic, Operational, Service-Orientated, Systems, Technical Standards, Acquisition, and All viewpoint). Each MODAF viewpoint is a suite of specific conceptual models (products). This work examines four viewpoints relevant to the study. A strategic viewpoint (StV) defines the desired target activity and identifies the capabilities needed to achieve that result. Operational Viewpoint (OV) defines the processes, information, and entities required to meet capability requirements. Service Orientated Viewpoint (SOV) defines the software services needed to support the processes described in the OV models. Systems Viewpoint (SV) defines application software systems – the physical implementation and solution of OV and SOV. The specifications of these mentioned MODAF viewpoints are analysed and a list of key concepts is provided in Table 1.

The newly created Unified Architecture Framework (UAF) is based on the Unified Profile for DoDAF and MODAF (UPDM) (OMG, 2017), (Hause *et al.*, 2016). UAF provides a set of rules to enable users to create consistent enterprise architecture (as a set of models) based on generic enterprise and system concepts with rich semantics (Morkevicius *et al.*, 2017). The Unified Architecture Framework is an Object Management Group (OMG) modelling standard. UAF is defined using the matrix ((Columns: *Taxonomy, Structure, Connectivity, Processes, States, Interaction scenarios, Information, Parameters, Constraints, Roadmap, Traceability*); Rows: *Metadata, Strategic, operational, Services, Personnel, Resources, Security, Projects, Standards, Actual Resources, Dictionary, Summary&Overview, Requirements*). “Along the left side of the matrix are the different levels of abstraction of the architecture: enterprise, service, logical, resources, deployed and architecture. Across the top of the matrix are the different types of diagram categories: classification, structure, connectivity, processes, states, sequences, information, constraints, and program. At the intersection of the matrices are the different views as well as a translation to the previous views for NAF and MODAF” (Hause *et al.*, 2016). The UAF does not define new conceptual structures (products), therefore UAF is a taxonomy that helps systematize the products (model types) of existing EA frameworks.

There is little work on the application of meta-models to EA frameworks. The work related to meta-modelling approach in EAF development is the OMG document of UAF Grid (UAF Domain Metamodel, 2020), including the layer “Meta-data Md” and specification of UAF domain meta-model elements. The “Meta-data Md” describes a metadata

Table 1
Summary of EA modelling concepts.

MODAF	MODAF concepts	Concepts of other approaches
Strategic viewpoint (StV)	Enterprise Vision, Whole Life Enterprise, Enterprise Goal, Enduring Task, Enterprise Phase, Capability, Capability Configuration, Capability Dependencies, Environment, Environment Property, Location, Exhibits, Actual Project, Actual Organisational Resource, Resource Interaction, Standard Operational Activity	Mission, Goal, Community, Company, Coordination, Artifacts, Task
Operational Viewpoint (OV)	Description, Location, Operational Activity, Capability, Problem Domain, Service, Node, Known Resource, Needle, Energy Flow, Material Flow, Information Element, Movement of People, Trustline, Logical Flow, Information Exchange, Operational Activity Flow, Operational Constraint, Operational State Description, Movement of people, Resource Interaction, Resource Type, Role Type, Organisational Resource, Post Type, Organisation type, Competence, Actual Organisation, Actual Post, Service, Process, Entity, Entity Relationship, Data Model, Attribute, Mission	Attributes, (Control effects), Gateways, Compensation, Transaction, Automated Transaction, Connecting Objects, Constraining Entity, Event, Rule, Meaning, Actor, Agent, ContAgent, Worker, Party
System viewpoint (SV)	Artifact, Organisational Resource (Organisation Type, PostType, RoleType), Software, Physical Architecture (CapabilityConfiguration, Service Implementation), ResourcePort, ResourceInterface, ResourceInteraction (Resource Person Flow, Resource Material Flow, Resource Energy Flow, Resource Communication)	Interface

layer that provides detailed definitions of EA views (a summary of, index to, the catalogs, matrices, diagrams) and serves as a common vocabulary for shaping EA decisions. These meta-data specifications do not include the EA design process, i.e. transformations within the elements of EA framework – between views, products of definite framework. Some works construct the meta-models of existing EA Frameworks, but do not extend them and do not relate to the analysis of reality domain characteristics (causal dependencies) and the impact on A solutions (Bernus and Noran, 2010). Therefore, meta-modelling approach in EAF development remains a relevant issue to be addressed.

A study on the structure of the EA frameworks (ArchiMate, DODAF, MODAF, UAF) and business process modelling approaches (BPMN, BPD, BMM, OSM, KAOS, MT, EMC) identified 133 different concepts in these techniques (Gelžinienė and Gudas, 2015). Analysis of the modelling methods reveals that MODAF covers most concepts used in other approaches and thus provides the most comprehensive framework for enterprise architecture development.

However, Table 1 shows that MODAF does not have certain concepts to indicate aspects that are taken into account in other modelling techniques for software development: *Collaboration*, *JobClassification*, *OrgAssignment*, *PosElement*, *ParticipationType*, *PosAssignment*, *PosAuthority*, *PosRequirement*, *RelationshipType*, *Representation*, *Addressable*, *ContactInfo*, *Coordination*, *Data processing (DA)*, *Decision*, *Decision implementation (SP)*, *Decision making (RE)*, *Primary data*, *Interpretation (IN)*.

Table 1 lists all MODAF concepts (second column) and is broken down into specific MODAF products (first column). The third column contains concepts from other methods,

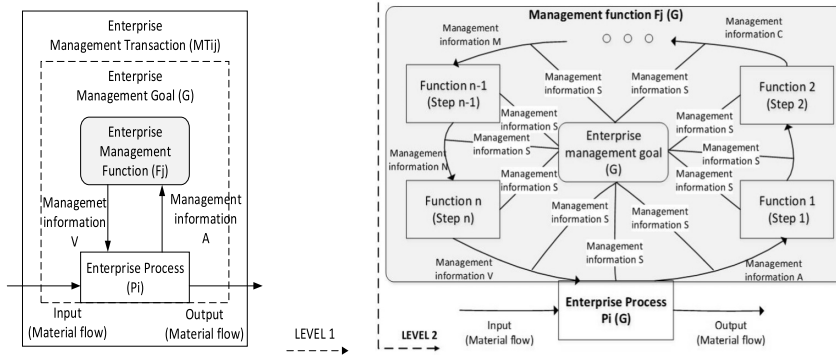


Fig. 3. Granularity of causal knowledge: Level 1 – Management Transaction, and Level 2 – Elementary Management Cycle frameworks.

meaningfully assigned to the relevant MODAF products. It is likely that MODAF can be supplemented by several meaningful concepts to improve the framework.

3. Subject Domain Causality Modelling

In a causal enterprise modelling approach, the management functional dependency (MFD) is defined as a primary cause that creates a causal behaviour between some subset of enterprise activities (a closed-loop chain of causal dependencies) (Gudas, 2012; Gudas and Lopata, 2016). MFD is a real-world phenomenon (causation) that is sought to be discovered by managers or domain analysts (or, in the case of incompetence, not realized). MFD predefines causal dependence of some activities (processes, operational capabilities or organizational units) required by particular business needs (i.e. strategic plan or actual business event). Perceived MFD is conceptualized as the Management Transaction (MT) and is described in detail as the Elementary Management Cycle (EMC) (Fig. 3).

Therefore, a two-level granularity of the domain causal knowledge can be achieved:

Level 1: MT framework reveals a higher level content of management activities: a closed-loop chain of information flows and transformations. In this approach, MT explores the first step of causal modelling of domain activities (i.e. this is conceptual representation of perceived MFDs). The conceptual structure of the management transaction is presented in Fig. 4: Pi – basic physical (material) process (i – identifier), Fj – management function (j – identifier), A – Process State Attributes (raw data), V – Controls (impacts to P).

Note that the enterprise goal G is not explicitly stated in the description of MT, only marked with a dotted line in Fig. 3, but it is considered by the analyst and affects the specification of MT.

Level 2: EMC framework reveals the internal structure of the MT framework as it decomposes the content of the management function Fj (G): internal steps of information transformations (functions) and information flows (A, B, ..., V) between steps (Fig. 3,

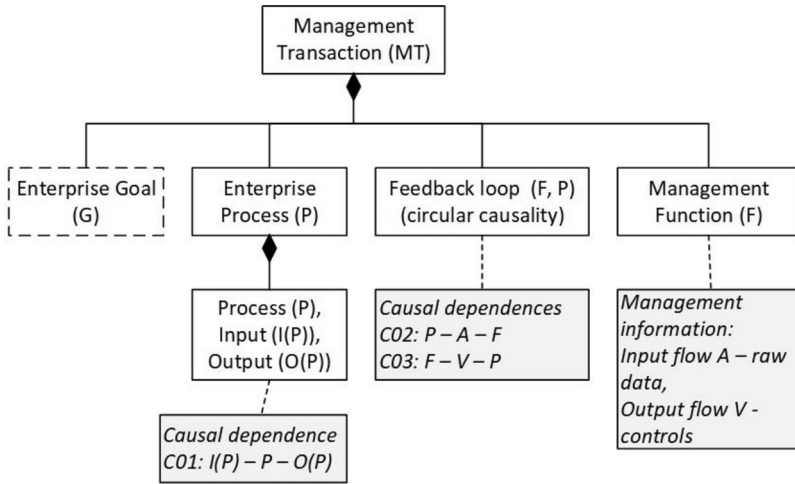


Fig. 4. Conceptual structure of Management Transaction.

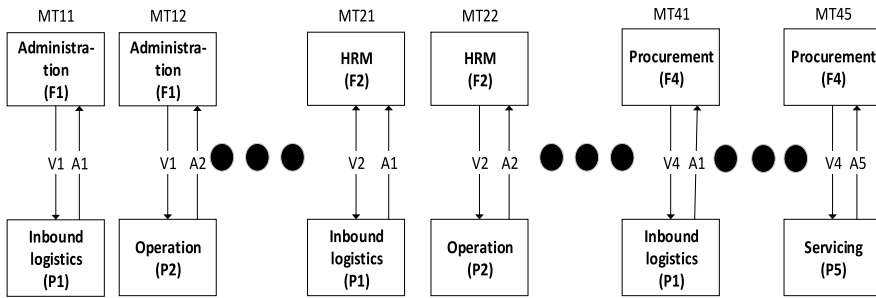


Fig. 5. Internal model of Porter's VCM as a system of management transactions.

Level 2), clearly indicate the management goal (G) and the impact (management information flows S) of G on the EMC components.

MT and EMC frameworks are unified components of causal knowledge (deep knowledge) for an enterprise causal model. EMC is an internal model of the Management Transaction (Gudas, 2012). A similar interpretation of the transaction as a complex component of deep knowledge (“molecule”, a unified building block) is described in the DEMO ontology (Dietz, 2006).

Therefore, a well-known business management model – Value Chain Model (Porter, 1985) is modified from the causal modelling viewpoint and depicted in Fig. 5 as a system of management transactions $\{MT_{11}, \dots, MT_{45}\}$:

- Support Activities are referred to here as *enterprise management functions* $F =$ (Administration (F1), HRM (F2), Finance Management (F3), Product and technology development (F4), and Procurement (F5));

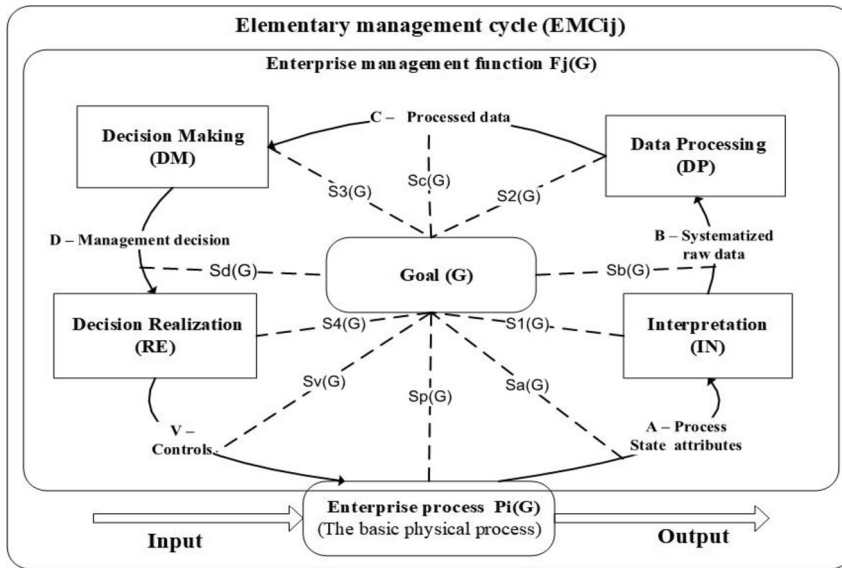


Fig. 6. Specific version of the EMC framework.

- Primary Activities are referred to here as *enterprise processes* $P =$ (Inbound Logistics (P1), Operation (P2), Outbound Logistics (P3), Sales and Marketing (P4), Servicing (P5)).

4. The Internal Structure of the Management Transactions

The general EMC framework in Fig. 3 (level 2) explicitly specifies the internal components of management transaction: steps (transformations) and interactions (information flows) of management function F . In order to reveal the content of the enterprise management information, a specific version of the EMC framework (Fig. 6) was developed (Gudas, 2012; Gudas and Lopata, 2016). This version of EMC includes components with well-defined semantics as follows: Goal (G), a goal-driven management function $F_j(G)$, enterprise process ($P_i(G)$), and connecting information flows (A, B, C, D , and V). Management function $F_j(G)$ is a complex structure, which consists of the *goal-driven procedural components* (four types of the information transformation steps IN, DP, DM , and RE) and the *management information flows* (A, B, C, D , and V denote the types of data/knowledge, and a flow type S denotes the impacts of goal G to other elements of EMC framework). The semantics of the *management information flows* is as follows: flow A is “process state attributes (raw data)”, flow B is “systematized raw data”, flow C is “processed data”, flow D is “management decisions”, and flow V is the “controls” of Process $P_i(G)$.

The semantics of the procedural components of EMC is as follows:

- *The Interpretation (IN)* performs the acquisition of raw data (of P process state) according to the needs of $F_j(G)$: identification, checking and systemizing of the required raw data A , according to the impact $S_1(G)$ of goal G .

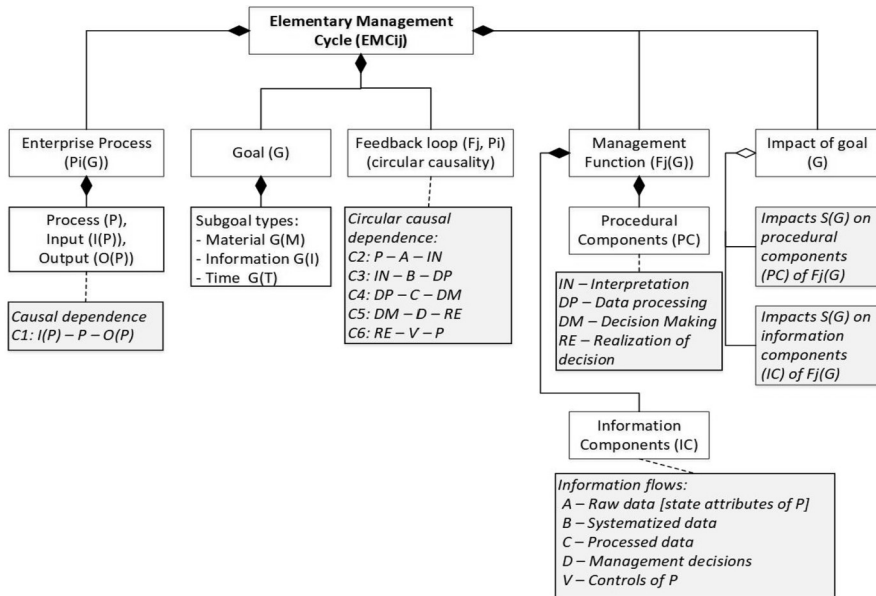


Fig. 7. Conceptual structure of EMC framework.

- *The Data processing (DP)* performs data transformations required for the content and task structure of the management function $F_j(G)$, and according to the impact $S_2(G)$ of goal G .
- *The Decision-making (DM)* generates management decisions based on the required content and task structure of the management function $F_j(G)$, and according to the impact $S_3(G)$ of goal G .
- *The Decision realization (RE)* accomplishes decisions required for the content and task structure of the management function $F_j(G)$, and according to the impact $S_4(G)$ of goal G .

Process $P_i(G)$ refers to the physical transformations that produce a tangible result of the enterprise. The procedural components (IN, DP, DM, and RE) denote steps of the feedback loop – circular causality between $P_i(G)$ and $F_j(G)$ (Fig. 7). Note, only the information content of the procedural components is considered here. Therefore, the content of IN, DP, DM, and RE refers to the knowledge clusters (Gudas, 2012, 2016):

- Interpretation (IN) is a cluster of knowledge (rules) for the collection, identification, and systematization of raw data;
- Data processing (DP) is a cluster of data processing knowledge;
- Decision making (DM) is a cluster of decision making knowledge (rules);
- Decision realization (RE) is a cluster of decision implementation knowledge;
- Goal (G) is a cluster of the enterprise strategy knowledge (requirements, constraints, capability specifications) that affect all other components of EMC (Fig. 7).

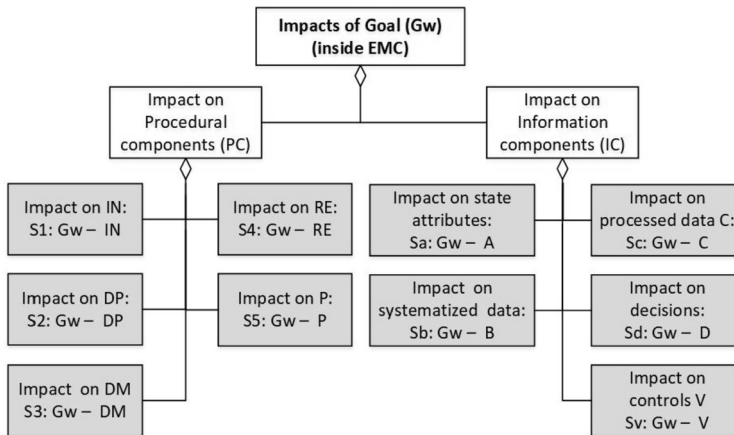


Fig. 8. Impacts of goal G on EMC components.

This comprehensive conceptual structure of EMC (Fig. 7) provides a systemic basis for reviewing the EA frameworks in terms of causal modelling.

An important part of domain causality modelling is to specify the impact of the goal G on other EMC components (procedural and information components). The detailed classification of the G impact on other EMC components is given in Fig. 8.

5. Causal Enterprise Architecture Modelling Constructs

The causal modelling paradigm is a priority of our approach, as the assumption is made that expert knowledge and decisions should be compared with discovered causal models of enterprise domain.

The analysis carried out suggests that the concept *Goal* in the context of EMC structure corresponds to co-related MODAF concepts *EnterpriseVision*, *EnterpriseGoal*, *Enduring-Task* and *Capability* of StV viewpoint. The concept *Capability* is key in the sense that it is *EnterpriseVision*, *EnterpriseGoal*, and *EnduringTask* driven concept and directly links StV products to OV, SV and SoV products and their concepts.

Examining the conceptual structure of EMC (Figs. 7 and 8) and comparing it with the MODAF meta-model, we found that some aspects of *Capability* dependence relationships and the impacts of enterprise goal are not included in the StV viewpoint. The conceptual structure of the EMC in Fig. 5 defines the following types of impacts of *Goal* on internal elements of any activity (management function), system or service:

- Impacts of *Goal* on procedural components of EMC defined here as management information flows S1, S2, S3, and S4; i.e. an impact of Goal on internal elements of operational activities (processes, actions, systems, services);
- Impacts of *Goal* on information components of EMC defined here as management information flows Sa, Sb, Sc, Sd, and Sv, i.e. an impact of Goal on internal information flows – inputs/outputs of operational activities (processes, actions, systems, services);

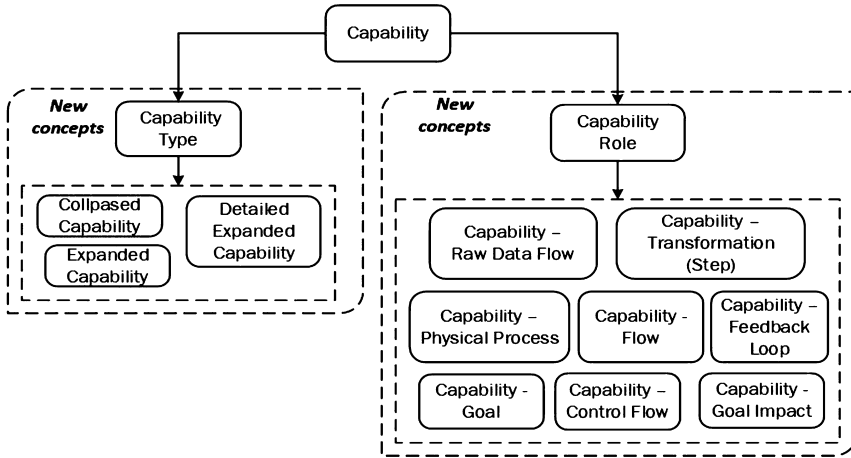


Fig. 9. Expanded concept *Capability* [in MODAF meta-model notation].

- Impacts of *Goal* on enterprise process *P* defined here as management information flow S5, i.e. an impact on physical processes – transformations of material flows or resources.

The requirement of taking into consideration the feedback loops between elements of any EA framework (capabilities, nodes or activities) is an essential pre-condition of well-managed enterprise processes. The origin of this requirement is the conceptual structures of the MT (Fig. 4) and EMC (Fig. 6).

Whereas the concept *Goal* (in the context of EMC) corresponds to the concept *Capability* (in the context of MODAF), on this basis, we propose more detailed modelling of the goal-related aspects in StV viewpoint (Fig. 8). Causality-based structure of the concept *Capability* should include a (lower level) predefined types (sub-capabilities): *Capability* of Information Components and *Capability* of Information Transformation Components. Both types of *Capability* (sub-capabilities) are applicable to all other MODAF products, namely OV, SV and SoV viewpoints. This allows us to specify more precisely the strategic view based requirements on elements of EA.

Causal modelling requires predicting the internal organization of processes. Therefore, we suggest predetermining potential causal dependencies in two competing ways: a) causal dependencies between identified capabilities in StV viewpoint and b) causal dependencies between identified operating nodes in OV viewpoint. Note that these techniques are complementary and can be used in parallel, simultaneously.

The definition of *Capability* has been expanded with new concepts aimed to define causal knowledge (Fig. 9):

- The expanded version of concept *Capability* includes *Capability Types* (*Collapsed Capability*, *Expanded Capability*, *Detailed Expanded Capability* and) and *Capability Roles* (*Capability-Flow*, *Capability-Raw-Data-Flow*, *Capability-Transformation (Step)*, *Capability-Control-Flow*, *Capability-Physical-Process*, *Capability-Goal*, *Capability-Goal-Impact*, and *Capability-Feedback-Loop*).

Table 2
Causal modelling concepts.

Capability types	Capability roles
C – Capability	C(P) – Physical process
CC – Collapsed Capability	C(S) – Information transformation (step)
CE – Expanded Capability	C(I) – Flow of information/knowledge [between C(S)]
CDE – Detailed Expanded Capability	C(V) – Control Flow [from C(I) to C(P)]
	C(A) – Raw Data Flow [from [C(P) to C(I)]
	C(G) – Goal of self-managed component
	C(D) – Goal Impact (dependence)
	C(FL) – Feedback loop (realized circular causality)

- *The Collapsed Capability* is a complex structure and consists of two Capability Types: *Expanded Capability* and *Detailed Expanded Capability*. A *Collapsed Capability* construct close to the concept of *Collapsed Sub-process* in BPMN 2.0.
- *The Expanded Capability* corresponds to the MT framework, which has a property of circular causality, and consequently, includes Capability Roles: *Capability-Raw-Data-Flow*, *Capability-Transformation*, *Capability-Physical-Process*, *Capability-Control-Flow*, and *Capability-Feedback-Loop*.
- *The Detailed Expanded Capability* corresponds to the EMC framework, which has a property of circular causality and consequently, includes Capability Roles: *Capability-Raw-Data-Flow*, *Capability-Flow*, *Capability-Control-Flow*, *Capability-Transformation (Step)*, *Capability-Goal*, *Capability-Goal-Impact*, *Capability-Physical-Process*, and *Capability-Feedback-Loop*.
- The *Capability* concept includes a new type of dependence – *Circular Causality Dependence*. This type of dependence is needed to specify the circular causality – the control loop between the physical process and information processing parts.

The introduced causal modelling concepts (Table 2) and the *Capability* meta-model in Fig. 10 provide the basis for developing causal meta-models of EA framework starting from StV and OV viewpoints.

The causal meta-model of *Capability* (Fig. 10) includes capability types *Collapsed Capability*, *Expanded Capability*, and *Detailed Expanded Capability*. The *Expanded Capability* (DE) includes *Capability Roles* that correspond to the MT framework. Therefore, *Expanded Capability* (CE) includes *Capability Roles* as follows: C(P), C(S), C(A), C(V) and C(FL). Capability role C(FL) must ensure that internal elements (i.e. *Capability Roles*) of CE are linked by circular causal dependence to form a feedback loop. The *Detailed Expanded Capability* (CDE) is of finer granularity and includes capability roles that correspond to the EMC framework. Therefore, *Detailed Expanded Capability* includes the Capability roles C(P), C(G), C(I), C(S), C(A), C(V) and C(FL). The dashed rectangle indicates that some *Capability-Step* C(S) may vary, depending on the inherent causality (regularity) of the specific domain. *Capability role C(FL)* must ensure that the internal elements (i.e. *Capability Roles*) of CDE are linked by circular causal dependence to form a feedback loop.

MODAF methodology provides a logical link between StV and OV viewpoints. The *Capabilities* identified by the StV viewpoint require the creation elements of *Operational*

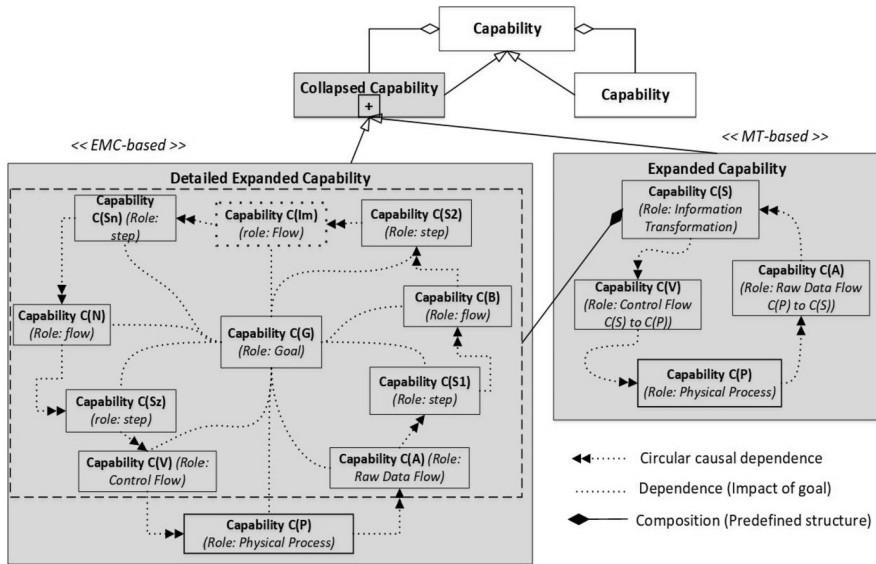


Fig. 10. Causal *Capability* meta-model.

Nodes of OV. The identified *Capability dependencies* are the basis for the *Operational Node Internal Relationship*.

Causal modelling approach considers *Operational Node* (i.e. a logical entity that performs operational activities) as a self-managed system relevant to Management Transaction (MT in Figs. 3 and 4). From here, it follows that a higher level *Operational Node* is considered as a complex component, which satisfies the circular causality requirement: any *Operation Node* in the next modelling step is decomposed as closed-loop interaction of operational activities. Therefore, any *OV* node includes at least two lower-level nodes (or activities) with an information feedback loop between them. *Note: In some simpler case OV, node could be considered as a transaction in terms of BPMN.*

Thus, the new types of causal *Operational Nodes* are defined as follows (Fig. 11):

- A *Collapsed Operational Node* is a complex node (assumed as a transaction). *Note:* Conceptually, this is close to the concept of *Collapsed Sub-process* BPMN 2.0;
- An *Expanded Operational Node* is defined as an MT-based framework: consists of Nodes (with predefined Roles *Raw-Data-Flow*, *Control-Flow*, *Information-Transformation*, and *Physical-Process*) and causal feedback in between. *Note:* conceptually, an *Expanded Operational Node* corresponds to some extent to the BPMN 2.0 *Expanded Sub-process* or *Transaction specification*;
- A *Detailed Expanded Operational Node* has a predefined structure relevant to the EMC framework: consists of several lower-level Nodes (with predefined Roles *Raw-Data-Flow*, *Control-Flow*, *Nodes-Steps*, and *Operational Nodes-Flows*) linked by circular causality dependence and including an *Operational Node-Goal*;
- *Note:* conceptually, a *Detailed Expanded Operational Node* is only to some extent compatible with the BPMN 2.0 complex *Transaction* specification.

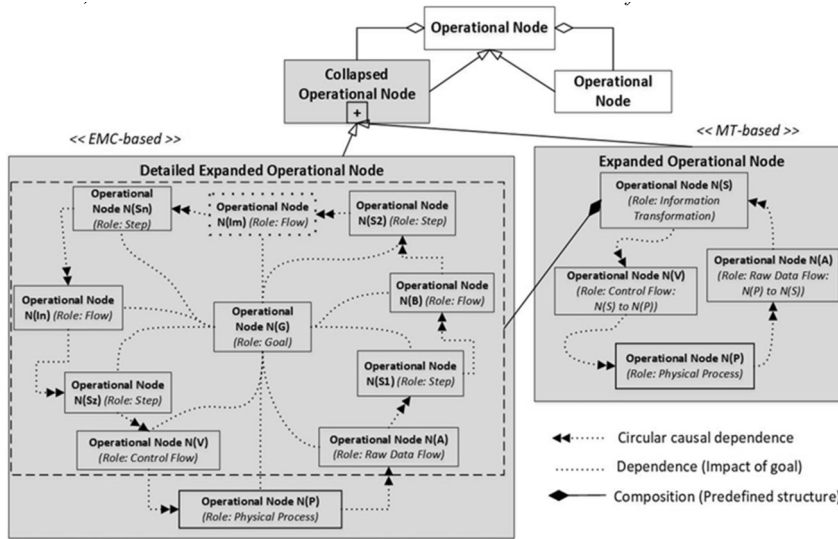


Fig. 11. Causal meta-model of Operational Node.

6. Causal models for Enterprise Architecture Development

6.1. The Causal Modelling in MDA/MDD Process

The traditional MDA scheme (CIM – PIM – PSM – Code) has been upgraded with causal modelling by adding a new layer of Domain Knowledge Model (DKM) in Gudas and Valatavičius (2020). The causality-driven MDA/MDD process (DKM – CIM – PIM – PSM – Code) starts with discovering of subject domain regularities (causation) and developing of Domain Knowledge Model (DKM). The proposed causality-driven EA development approach is aligned with the modified (causal) MDA/MDD process in Fig. 12.

Prerequisites for the development of the causal knowledge-based EA solutions:

Assumption 1. A Domain Knowledge Model (DKM) layer of the modified MDA/MDD process is for discovering subject domain regularities (causal knowledge) and developing a domain knowledge model (DKM).

In the next step, the DKM is transformed into causal metamodels of the selected EA system.

Depending on the type of domain under consideration, different patterns of causality can be identified. Domain Knowledge Model (DKM) is an internal model of reality domain that describes domain-specific causality (system of causal dependencies, regularities). From the causal enterprise modelling perspective (Gudas, 2012, 2016), two circular causality patterns have been singled out, they are of different granularity: the first is the Management Transaction (MT) framework, and the second, a more detailed one, is Elementary Management Cycle (EMC) framework. The DKM is transformed into causal

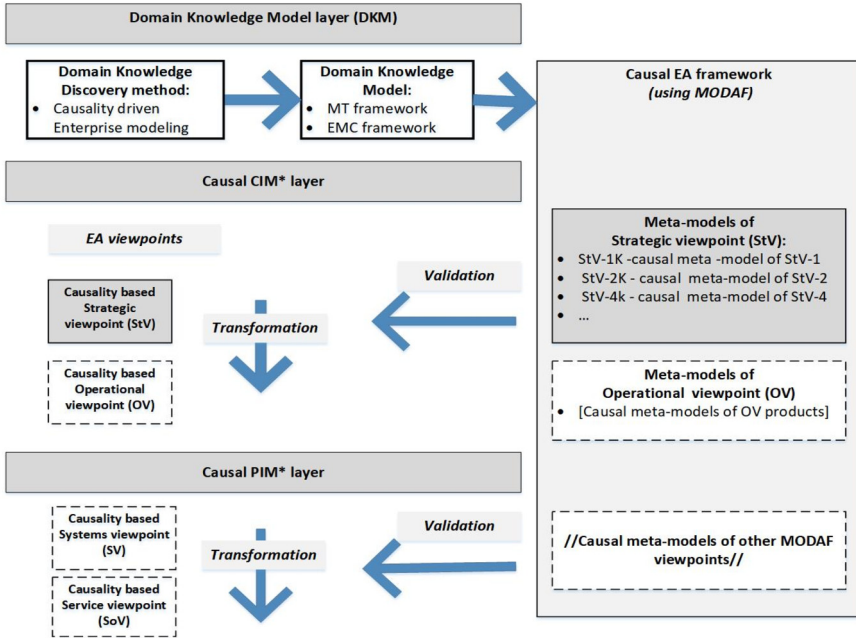


Fig. 12. Causality-driven EA development (using MODAF framework).

meta-models of the selected EA framework. Causal meta-models are used as a priori (scientific) knowledge (mandatory constraints) to validate specific EA models.

Assumption 2. Starting with EA development, all empirically identified Capabilities are defined as Collapsed Capability. The meta-models of Collapsed Capability, Expanded Capability, and Detailed Expanded Capability are key causal knowledge items for developing causality-based enterprise architecture.

In the course of causal EA development, the *Collapsed Capability* is mapped to the *Expanded Capability* or *Detailed Expanded Capability*. The Domain Knowledge Model predefines the internal structure of the *Collapsed Capability*, i.e. predefines the meta-models of *Expanded Capability* and *Detailed Expanded Capability*. In our approach, the internal structure of the *Expanded Capability* matches the MT framework and the internal structure of the *Detailed Expanded Capability* matches the EMC framework.

Suppose the CIM layer comprises strategic and operational viewpoints of MODAF (in our case study). Strategic viewpoint (StV) is depicted as a hierarchy of the key concepts *A Whole Life Enterprise*, *Enterprise Phase*, *Collapsed Capability* (1), *Collapsed Capability* (2) and *Expanded Capability*. A top-level concept “A Whole Life Enterprise” (WLE) consists of *Enterprise Phases* (EF) linked by causality relations. The concept “Enterprise Phase” (EF) is considered as a white box, the internal structure of EF consists of “*Collapsed Capabilities*” (CC) with the internal structure corresponding to the Domain Knowledge Model (DKM).

Table 3
Concepts of the Domain Knowledge Model (DKM).

MT framework	MT – Management Transaction; P – Enterprise process, F – Enterprise management function, A – Process state attributes (raw data), V – Controls (impacts to P), G – Enterprise goal (not specified explicitly), FL – Feedback loop between P and F
EMC framework	EMC – Elementary Management Cycle, defines as a detailed structure of MT; P – Enterprise process, F – Enterprise management function, A – Process state attribute flow, V – Controls flow (impacts to P), B – Systematized raw data, C – Processed data flow, D – Management decision flow, IN – Interpretation activity, DP – Data processing activity, DM – Decision-making activity, RE – Decision realization activity (implementation), G – Enterprise goal (explicitly specified), C2 – Causal relation of P to IN via flow A, C3 – Causal relation of IN to DP via flow B, C4 – Causal relation of DP to DM via flow C, C5 – Causal relation of DM to RE via flow D, C6 – Causal relation of RE to P via flow V, G(IC) – Goal impact to information components (IC), G(PC) – Goal impact to procedural components (PC)

Assumption 3. *The role of EA meta-models is twofold: (a) validation of the causal dependencies of vertical transformations (top-down decomposition in the MDA/MDD process) and (b) validation of causal dependencies at each level of EA development using some selected EA framework (EA methodology).*

In our case study, the MT framework and EMC framework represent the causal knowledge structures, and they are used to specify the causal meta-models of EA viewpoints. The causal EA meta-models are the basis to: a) reveal a set of the *Capability Types and Capability Roles*, and b) validate the causal dependencies between the lower-level elements of the *Collapsed Capability* (see Fig. 10).

The causal development of EA using the MODAF is consistent with the causal-based MDA/MDD process, as shown in Fig. 12. The theoretical foundations of knowledge discovery in the field of enterprise application engineering are described in more detail (Gudas, 2012, 2016). Based on this, a domain knowledge model (DKM) is developed and applied to the causal approach to EA development.

The DKM concepts in Table 3 refer to the types of activities and flows that are specific to the causal dependencies of the enterprise domain. *Note: In general, the number of steps in the EMC framework is undefined, as shown in Fig. 3.* In the example below, DKM includes a specialized EMC framework (Figs. 6 and 7) with four well-defined steps (IN, DP, DM, and RE) (Gudas, 2012).

The concepts in Table 1 are used in the Strategic viewpoint (StV) meta-models. We can say that concepts of the StV meta-models denote clusters of causal knowledge of the domain, i.e. as defined by DKM capability types and capability roles. The rules for transformation of the Domain Knowledge Model (DKM) to StV viewpoint meta-models are presented in Table 4.

The development of an EA solution under MODAF starts from the StV viewpoint development.

The expert conducts an analysis of the strategic perspective, the purpose of which is to build a set of *Capabilities* required to implement the enterprise strategy. Identified *Capabilities* are specified in the StV-1 model and guide further EA solutions as *Capabilities* are mapped to OV and other viewpoints.

Table 4
Rule for DKM mapping to StV viewpoint meta-models.

Mapping Rules (MR)	Description of mapping rules
<i>StV meta-model capability types discovery rules</i>	<i>StV meta-model corresponds to the MT framework or EMC framework</i>
MR01: MT → CC → CE	Management Transaction (MT) corresponds to Capability type CC (<i>Collapsed Capability</i>), and CC is defined as <i>Expanded Capability</i> (CE)
MR02: EMC → CC → DCE	Elementary Management Cycle (EMC) corresponds to Capability type CC (<i>Collapsed Capability</i>), and CC is defined as <i>Detailed Expanded Capability</i> (CE)
<i>CE internal model discovery rules</i>	<i>An internal model of the Expanded Capability (CE) corresponds to the MT framework</i>
MR03: P → C(P)	Capability role C(P) corresponds to the physical process (P)
MR04: F → C(S)	Capability role C(S)) corresponds to the management function F
MR05: A → C(A)	Capability role C(A) corresponds to the raw data flow from P to F
MR06: V → C(V)	Capability role C(V)) corresponds to control flow from C(S) to C(P)
<i>DCE internal model discovery rules</i>	<i>An internal model of the Detailed Expanded Capability (CDE) corresponds to the EMC framework</i>
MR03: P → C(P)	Capability role C(P) corresponds to the physical process (P)
MR04: F → C(S)	Capability role C(S)) corresponds to the management function F
MR05: A → C(A)	Capability role C(A) corresponds to the raw data flow from process P to F
MR06: V → C(V)	Capability role C(V)) corresponds to the control flow from F to P
MR07: B → C(B)	Capability role C(B) corresponds to the information flow between steps IN and DP of EMC
MR08: C → C(C)-	Capability role C(C) corresponds to the information flow between steps DP and DM of EMC
MR09: D → C(D)-	Capability role C(N) corresponds to the information flow between steps of EMC
MR10: IN → C(IN)	Capability role C(IN) – corresponds to the Interpretation activity (IN)
MR11: DP → C(DP)	Capability role C(DP) corresponds to the Data processing (DP) activity
MR12: DM → C(DM)	Capability role C(DM) corresponds to the Decision making (DM) activity
MR13: RE → C(RE)	Capability role C(RE) corresponds to the Decision realization activity (RE)
MR14: G(PC) → C(PC)	Capability role C(PC) corresponds to the Impact of Goal to Procedural Components (PC) of EMC
MR15: G(IC) → C(IC)	Capability role C(IC) corresponds to the Impact of goal to Information Components (IC) of EMC

6.2. EA Development Tool with Knowledge Base

Enterprise Architecture (EA) tool is a development environment designed to store and present information related to EA solutions. Current EA development tools visualize the created EA models, help check model syntax, but cannot perform semantic analysis (validation) of EA solution, as have no domain knowledge models.

The alignment of the EA development and MDA/MDD process based on causal models reveals the structure and architecture of intelligent EA tools. Causal domain knowledge is predefined knowledge and at the engineering level is specified in the knowledge base using meta-modelling technique.

Creating a knowledge base is an intellectual work done by a domain analyst with scientific (pre-acquired) knowledge about the causality of a particular field and an expert in particular EA framework.

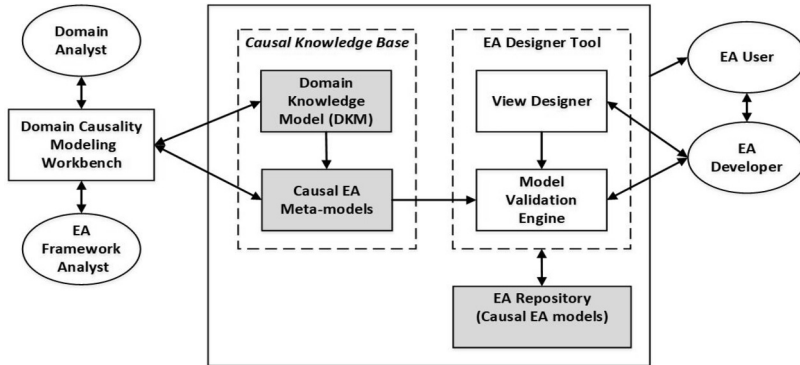


Fig. 13. EA development tool architecture with causal knowledge base.

Figure 13 shows EA development tool architecture improved using Causal Knowledge Base and model validation component. The Causal Knowledge Base consists of two parts: Domain Knowledge Model (DKM) and causal meta-models that support specific EA frameworks (e.g. MODAF, ArchiMate, etc.).

In creating a domain-specific EA project, the knowledge base is an active participant in the EA development process, a source of enterprise knowledge along with the EA developer and EA user. The knowledge base is a source of predefined causal knowledge because EA meta-models are used to validate EA developer decisions by validating developed EA models against the meta-models. As an example, rules for validating of the MODAF StV viewpoint are presented in Table 5. The Domain Knowledge Model and EA meta-models, together with the appropriate algorithms, provide a new opportunity to test and validate EA development solution, ensuring the consistency of the EA project.

6.3. Causality-Based EA Development

The first steps of causality-based EA development are very different, qualitatively different from traditional one.

The EA development stages on the causal CIM* layer cover these crucial moments:

1. The initial set of identified capabilities is the result of the agreement and observation-based analysis of enterprise vision, strategies, goals and enduring tasks. The (obtained, determined) identified capabilities are drawn up empirically since no formal methods or domain knowledge models are not foreseen for validation.
2. Causal development method describes each identified capability as *Collapsed Capability* (Table 4, mapping rules MR01 and MR02). In the next step, each *Collapsed Capability* is decomposed to determine its internal model, which is the *Expanded Capability* (Table 4, mapping rules MR03–MR06) or/and *Detailed Expanded Capability* (Table 4, mapping rules MR03–MR15).
3. Causality-based development of the EA solutions starts by validation of the internal structure of identified *Collapsed Capabilities* against the meta-model of the *Expanded*

Table 5
StV viewpoint validation rules.

Validation Rules (VR)	Description
Capability type validation	
VT01: $\{C\} \rightarrow \{CC\} \neq \emptyset$	Observation-based capability (C) is marked as Collapsed Capability (CC)
VT02: $\{CC\} \rightarrow \{CE, CDE\} \neq \emptyset$	An internal model of the Collapsed Capability (CC) is defined as <i>Expanded Capability</i> (CE) or/and <i>Detailed Expanded Capability</i> (CDE)
Capability role validation	
MT-based validation rules	
VR01: $C(P) \rightarrow \{E(C(P))\} \neq \emptyset$	There is at least one StV element E with a role C(P): $\{E(C(P))\} \neq \emptyset$, $\{E(C(P))\}$ not an empty set
VR02: $C(S) \rightarrow \{E(C(S))\} \neq \emptyset$	There is at least one StV element E with a role C(S): $\{E(C(S))\} \neq \emptyset$, $\{E(C(S))\}$ not an empty set
VR03: $C(A) \rightarrow \{E(C(A))\} \neq \emptyset$	There is at least one StV element E with a role C(A): $\{E(C(A))\} \neq \emptyset$, $\{E(C(A))\}$ not an empty set
VR04: $C(V) \rightarrow \{E(C(V))\} \neq \emptyset$	There is at least one StV element E with a role C(V): $\{E(C(V))\} \neq \emptyset$, $\{E(C(V))\}$ not an empty set
EMC-based validation rules	
VR01: $C(P) \rightarrow \{E(C(P))\} \neq \emptyset$	There is at least one StV element E with a role C(P), i.e. $\{E(C(P))\}$ not an empty set
VR02: $C(S) \rightarrow \{E(C(S))\} \neq \emptyset$	There is at least one StV element E with a role C(S), i.e. $\{E(C(S))\}$ not an empty set
VR03: $C(A) \rightarrow \{E(C(A))\} \neq \emptyset$	There is at least one StV element E with a role C(A), i.e. $\{E(C(A))\}$ not an empty set
VR04: $C(V) \rightarrow \{E(C(V))\} \neq \emptyset$	There is at least one StV element E with a role C(V), i.e. $\{E(C(V))\}$ not an empty set
VR05: $C(B) \rightarrow \{E(C(B))\} \neq \emptyset$	There is at least one StV element E with a role C(B), i.e. $\{E(C(B))\}$ not an empty set
VR06: $C(C) \rightarrow \{E(C(B))\} \neq \emptyset$	There is at least one StV element E with a role C(C), i.e. $\{E(C(C))\}$ not an empty set
VR07: $C(D) \rightarrow \{E(C(D))\} \neq \emptyset$	There is at least one StV element E with a role C(D), i.e. $\{E(C(D))\}$ not an empty set
VR08: $C(IN) \rightarrow \{E(C(IN))\} \neq \emptyset$	There is at least one StV element E with a role C(IN), i.e. $\{E(C(IN))\}$ not an empty set
VR09: $C(DP) \rightarrow \{E(C(DP))\} \neq \emptyset$	There is at least one StV element E with a role C(DP), i.e. $\{E(C(DP))\}$ not an empty set
VR10: $C(DM) \rightarrow \{E(C(DM))\} \neq \emptyset$	There is at least one StV element E with a role C(DM), i.e. $\{E(C(DM))\}$ not an empty set
VR11: $C(RE) \rightarrow \{E(C(RE))\} \neq \emptyset$	There is at least one StV element E with a role C(RE), i.e. $\{E(C(RE))\}$ not an empty set
VR12: $C(PC) \rightarrow \{E(C(PC))\} \neq \emptyset$	There is at least one StV element E with a role C(PC), i.e. $\{E(C(PC))\}$ not an empty set
VR13: $C(IC) \rightarrow \{E(C(IC))\} \neq \emptyset$	There is at least one StV element E with a role C(IC), i.e. $\{E(C(IC))\}$ not an empty set

Capability (Table 5, validation rules VR01–VR04) or/and *Detailed Expanded Capability* (Table 5, validation rules VR01–VR13).

Thus, there are two possible types of validation based on causal knowledge:

- a) Validation (1): Assessment of the correspondence of internal structure of *Collapsed Capabilities* against causal meta-model of Expanded Capability or Detailed Expanded Capability;

- b) Validation (2): Discovery of the dependencies between identified (observation-based) *Collapsed Capabilities* using the causal meta-model of *Expanded Capability* or *Detailed Expanded Capability*.

Causal EA development illustrated here using the MODAF framework and case study “Search and Rescue Enterprise (SAR)” (MODAF Strategic Viewpoint, 2019). Due to the limited scope of the article, we provide only a few additions of the MODAF StV viewpoint. Causality-based add-ons of MODAF include meta-models StV-1K, StV-2K, and StV-4K of Strategic viewpoint (StV) products StV-1, StV-2 and StV-4.

In the given example, the meta-models are considered as requirements (semantic constraints) for the development and validation of StV-1, StV-2 and StV-4 developed for a particular enterprise. Semantic constraints here mean predefined internal structure of the *Collapsed Capability*: a set of required capability types and causal dependencies inside capability types *Expanded Capability* and *Detailed Expanded Capability* (see Figs. 8, 10 and 11).

7. Domain Knowledge-Based Add-Ons for StV Viewpoint

New concepts related to causal knowledge have been included in StV’s viewpoint products as follows: Enterprise Vision (StV-1), Capability Taxonomy (StV-2) and Capability Dependence (StV-4). The Enterprise Vision StV-1 of MODAF provides an observation-based list of *Capabilities* (Enterprise Phase related). This is empirical information whereas MODAF makes no formal or conceptual requirements for the content of the capabilities identified in the development of the enterprise vision. The next stage is mapping the StV-1 to other StV viewpoint products (StV-2, StV-4, etc.), and thereafter, further EA development by mapping the identified set of capabilities to the OV viewpoint.

The causality-based Enterprise Vision StV-1 development is based on the causal meta-model StV-1K (Fig. 14). The causality-based Enterprise Vision meta-model StV-1K includes an observation-based list of concepts (the upper part of Fig. 14) and Domain Knowledge Model-based a structure of causal concepts (the lower part of Fig. 14).

The development of the causal StV-1 model also starts with an initial list of capabilities (empirical information). Let us assume that all identified capabilities are declared as *Collapsed Capabilities* (by definition in Assumption 2) with a predefined internal structure. The internal structure of identified *Collapsed Capabilities* needs to be refined in the next step using meta-models StV-2K (Fig. 15) and StV-4K (Fig. 16). There are two options to specify the internal structure of the *Collapsed Capability*: selecting capability type *Expanded Capability* or/and *Detailed Expanded Capability*.

Uncertainty arises in the development of the MODAF causal dependence model StV-4 – what objective evidence confirms the dependencies between the capabilities? Of course, the dependence of StV-4 capabilities is based on expert knowledge, experience, and opinion, but this is not consistent with the MDD methodology, as a transformation from a previously developed model would be required.

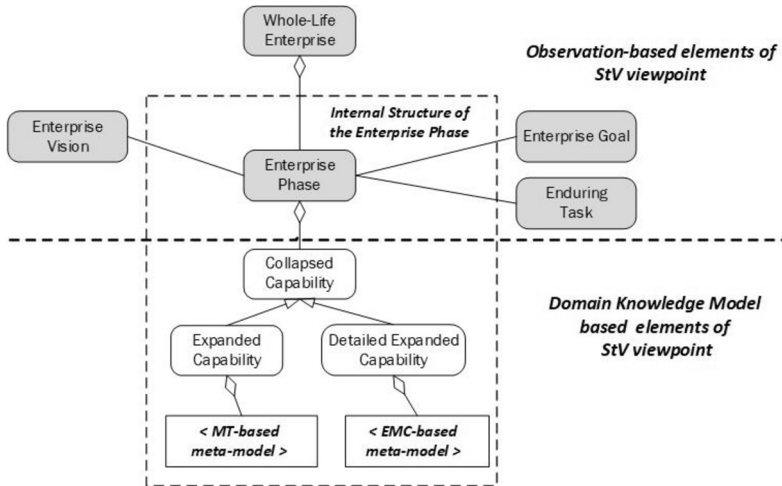


Fig. 14. Causal meta-model StV-1K of the Enterprise Vision [MODAF meta-model tagging].

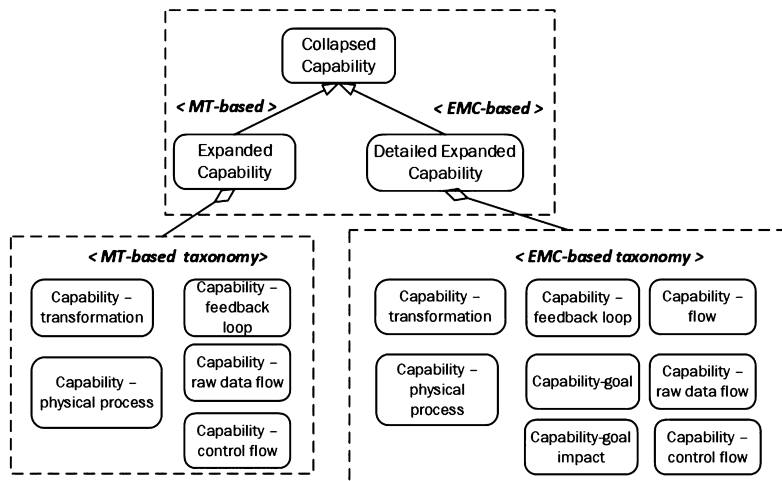


Fig. 15. Causal Capability Taxonomy meta-model StV-2K [MODAF meta-model tagging].

The causal Capability Taxonomy meta-model StV-2K depicts the required capability roles (Fig. 14). The Capability Causal Dependence meta-model StV-4K depicts required causal dependencies between capability roles (Fig. 16). The StV-2K is logically linked to StV-4K because the set of all identified in the StV-2K capabilities must be associated with causal dependencies as predefined in the StV-4K. StV-4K includes two (alternative) capability dependence structures of different granularity: MT-based and EMC-based capability dependence.

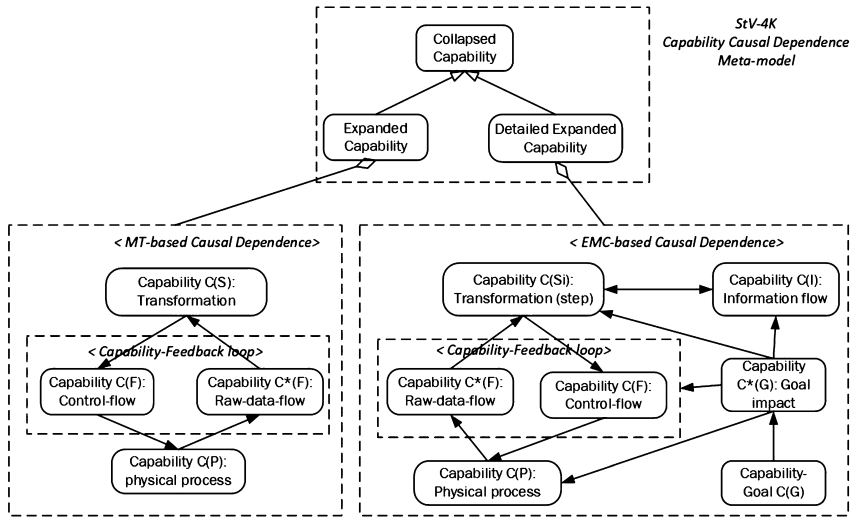


Fig. 16. Capability Causal Dependence meta-model StV-4K [MODAF meta-model tagging].

8. A Case Study of Causal StV Viewpoint Development

Let us take the case study “Search and Rescue Enterprise (SAR Enterprise)” (MODAF Strategic Viewpoint, 2019) as a starting point to explain a causality-based approach. In the original example (as well as in Fig. 17), the capability SAR is an aggregate element, it includes three components: capabilities Recovery, Search and Assistance. As well capability SAR is a generalized item: it provides two types of services: Capability Land SAR and Capability Maritime SAR.

The DKM layer distinguishes two main activities: to understand the knowledge discovery method and apply it to the development of the domain knowledge model. The main concepts of the traditional StV viewpoint are depicted on the left side of Fig. 17. Causal modelling concepts (i.e. meta-models of EA) are depicted on the right side and lower part of Fig. 17. Four levels of hierarchy from a *Whole Life Enterprise* to *Collapsed Capability* (2) are obtained from observation-based experience. However, this empirical solution needs to be validated. Causal EA modelling starts at the *Expanded Capability* level as this is where the internal model of *Collapsed Capability* (2) (i.e. meta-model) is selected for validation in the next stages of EA development. The following are examples of *SAR Enterprise* causal modelling, focused on the detailing Search capability. Similarly, Rescue and Assistance capabilities causal modelling should be done, creating meaningful causal structures of StV products that match the DKM.

Development of the causal StV solution (see Fig. 12) is a two-dimensional process:

- Vertical direction: transformations of StV models (StV-1, StV-2, and StV-4) in four levels of hierarchy follow to the MODAF methodology and
- Horizontal direction: causality validation process of internal structure of StV models (using Validation Rules in Table 5).

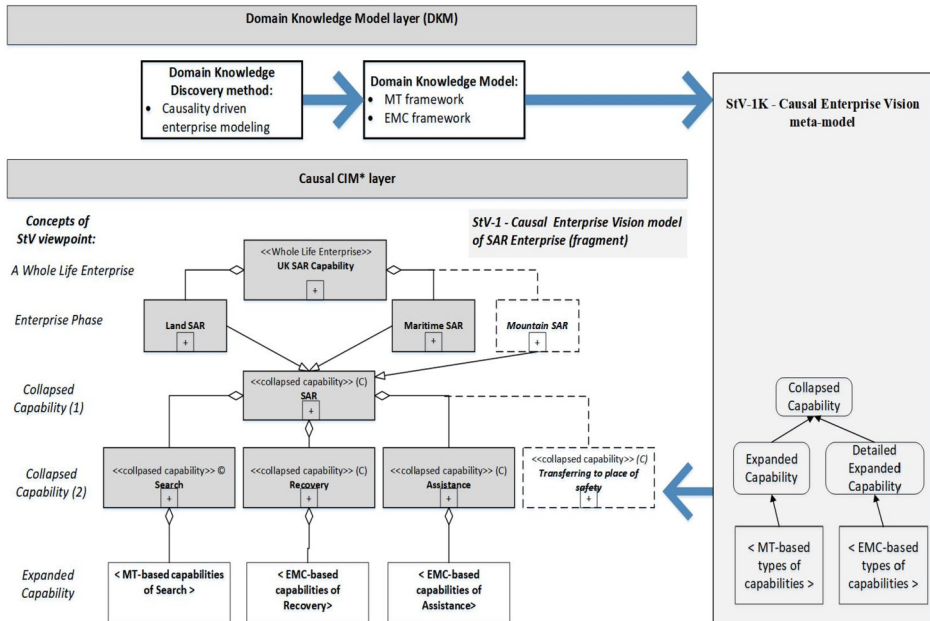


Fig. 17. Causal Enterprise Vision StV-1 of the SAR Enterprise (DKM based).

Let us review the main steps of causal validation of the Enterprise Vision StV-1 model:

- Capability type validation rule VT01: All identified capabilities of SAR Enterprise are under review; at least one must be indicated as *Collapsed Capability*.

Following the MODAF methodology, UK SAR Capability is assigned to the level of a *Whole Life Enterprise*. On the level of *Enterprise Phase* where are two types of *Collapsed Capability* Land SAR, and Maritime SAR. Let us assume that Land SAR and Maritime SAR are two parallel *Enterprise Phases*, in other words, these are two types of SAR enterprise (specializations). On the level of *Collapsed Capability (1)* depicted generic *Collapsed Capability* SAR consists of two capability types Land SAR and Maritime SAR.

- Capability type validation rule VT02: At the *Collapsed Capability (2)* level the internal structure of *Collapsed Capabilities* is assigned (i.e. CC is mapped to CE or/and CDE). Let us say, that EA developer (architect) has considered the *Collapsed Capabilities* of SAR Enterprise as follows: *Search* will be specified as (MT-based) *Expanded Capability (CE)*, *Recovery* and *Assistance* – as (EMC-based) *Detailed Expanded Capabilities (CDE)*.

Basic steps in the development of causal capability taxonomy StV-2 model:

- Capability role validation rules (VR01–VR04) or/and rules (VR03–VR15) used to specify a causal capability taxonomy StV-2 model in Fig. 18. Note, that the original capability taxonomy model StV-2 of SAR Enterprise was developed on an empirical

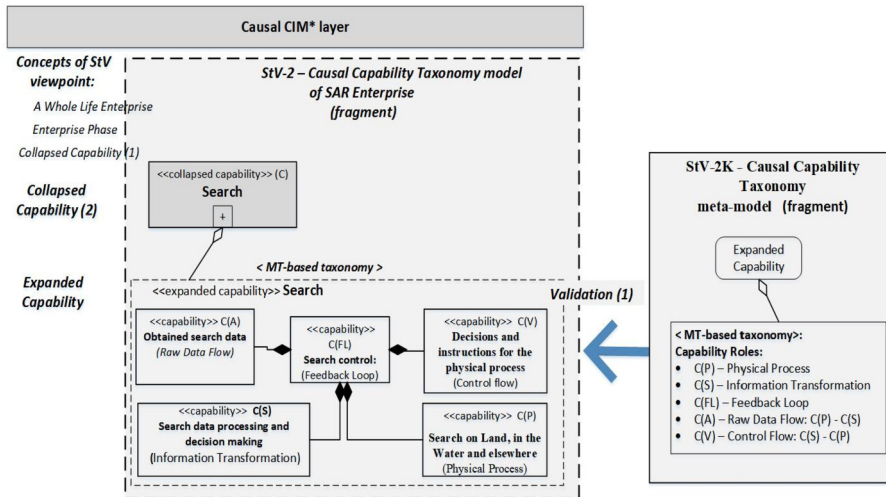


Fig. 18. SAR Enterprise: Causal Capability Taxonomy StV-2 model (fragment: shows *Expanded Capability* level of Search in detail).

(expert) knowledge basis. The causal EA modelling results in a more reasonable StV-2 due to validation using causal knowledge, fixed in the meta-model StV-2K.

- A twofold validation of the internal structure of *Collapsed Capability*:
 - Capability role validation (1): set of rules (VR01–VR04) evaluate the correspondence of the specified instance (on the level of *Expanded Capability*) to the meta-model of the *Expanded Capability* (Fig. 10);
 - Capability role validation (2): set of rules (VR01–VR13) evaluate the correspondence of the specified instance (on the level of *Expanded Capability*) to the meta-model of the *Detailed Expanded Capability* (Fig. 10).
- A twofold validation of the causal dependencies between capabilities on the level *Collapsed Capability* (2):
 - Capability role validation (3): set of rules (VR01–VR04) evaluate the correspondence of the causal dependencies between capabilities on the level *Collapsed Capability* to the meta-model of the *Expanded Capability* (Fig. 10);
 - Capability role validation (4): set of rules (VR01–VR13) evaluate the correspondence of the causal dependencies between capabilities on the level *Collapsed Capability* to the meta-model of the *Detailed Expanded Capability* (Fig. 10).

An example of the capability role validation (1). For instance, since capability Search in the previous stage (rule VT02) was specified as the *Expanded Capability*, the MT-based capability role validation rules (VR01–VR04) are applied to verify the current state of Search model (level *Expanded Capability* in Fig. 17). Therefore, causal model of Search must include sub-capabilities as follows (Fig. 18): *C(P)*: the physical process (to find the victim on the land, in the water and elsewhere), *C(S)*: Search data processing and decision making, *C(FL)*: Search Control (feedback loop), *C(A)*: Obtained data flow, *C(V)*: Flow

of decisions and instructions to the physical process $C(P)$. Note: Fig. 18 shows only a fragment of StV-1, *Expanded Capability* level of Search on Causal CIM* layer.

Similarly, *Collapsed Capabilities Assistance and Recovery* (in the previous stage (rule VT02) were specified as the *Detailed Expanded Capability*) should be decomposed in this way and validated using the EMC-based capability role validation rules (VR01–VR13).

An example of the capability role validation (3). Suppose that the capabilities *Search, Assistance and Recovery* are found to be causally related and should be consistent with the MT framework, so their interdependencies correspond to capability type *Expanded Capability*:

- VR01: Search is marked as the capability role $C(P)$,
- VR02: Recovery is marked as the capability role $C(S)$,
- VR03: Assistance is marked as the capability role $C(A)$,
- VR04: A capability role $C(V)$ is an empty set,
- Causal dependencies are consistent with the MT framework and associate them as follows:

Search ($C(P)$) → Recovery ($C(S)$) → Assistance ($C(A)$) → (*not defined* ($C(V)$)),

- Conclusion: The requirement of the circular causality is not satisfied at the level *Collapsed Capabilities (1)* of StV-1 since VR04 failed.
- Consequently, there is a logical gap at the level *Collapsed Capabilities (1)* of StV-1 – some required capability type $C(V)$ is missing. Therefore, some new activity of SAR Enterprise is required in this version of the StV-1 model.
- Suppose the proposed new capability is “Transferring to the place of safety” (capability type $C(V)$) that creates an MT-based circular causality structure at the level *Collapsed Capability (1)* (Fig. 18):

Search ($C(P)$) → Recovery ($C(S)$) → Assistance ($C(A)$)
→ Transferring to the place of safety ($C(V)$).

Note. A possible alternative if there is no causal dependency between *Search, Assistance, and Recovery* (activities occur in parallel), that is, are autonomous activities.

Now let us discuss the peculiarities of causal modelling on the level *Enterprise Phases*. The two *Collapsed Capabilities* Land SAR and Maritime SAR are required parts of *Whole Life Enterprise UK SAR*, and (at the same time) are specialization (sub-types) of the capability SAR depicted at the level *Collapsed Capability (1)*. Suppose the rules (VR01–VR04) were applied for the capabilities Land SAR and Maritime SAR (Fig. 17) on the level Enterprise Phase, this validation revealed a gap, and the new capability “Mountain SAR” was proposed.

Causal modelling of capabilities *Land SAR* and *Maritime SAR* can be performed according to the example with capabilities *Search, Assistance and Recovery*. Land SAR and Maritime SAR activities are tailored to the different real-world domains (land, water or

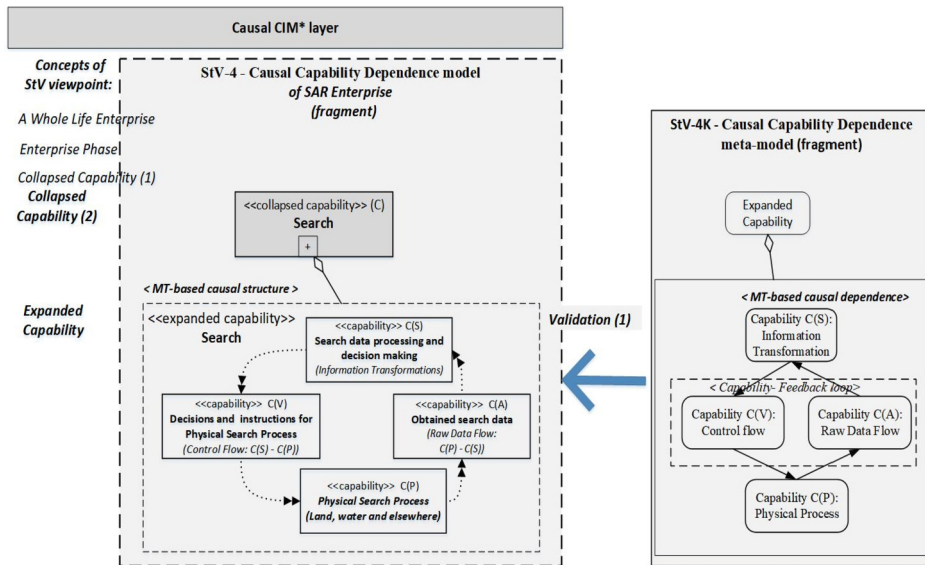


Fig. 19. Causal capability dependence model StV-4 of SAR Enterprise (fragment: shows the internal model of the Search).

elsewhere) respectively. The specificity of Land SAR and Maritime SAR activities are captured in specifications StV-1, StV-2, and StV-4.

In the next step, the causal dependencies are determined to develop the StV-4 model. The capability dependence model (StV-4) of the case study “SAR Enterprise“ (MODAF Strategic Viewpoint, 2019) has been developed on an observation basis as well. The causal EA approach can result in a more reasonable StV-4 product where the identified capabilities and dependencies between capabilities are verifiable by causal knowledge, fixed in the StV-4K meta-model.

Key steps in the development of causal capability dependence StV-4 model of the SAR Enterprise:

- Suppose that an internal structure of the capability *Search* is defined by MT-based meta-model, therefore rules (VR03–VR06) were used to form a causal capability taxonomy StV-4 model (Fig. 19). The lower level capabilities of the *Expanded Capability Search* are classified as follows: C(P) – the physical process (distress signal monitoring, find victim), C(A) – the raw data flow of obtained information (distress signal and related data), C(S) – data processing and decision making (distress signal processing, warning order sending, and health monitoring data), C(V) – feedback control flow of decisions and instructions (control impacts) for the physical process.
- Thus, the causal approach assures the consistent modelling – StV-2 transformation to StV-4 is based on the causal knowledge structures – StV-2K and StV-4K meta-models.

Causal enterprise architecture modelling begins by revealing the subject domain causation (regularities) and building the domain knowledge model (DKM) upon them. In our

previous work, the traditional MDA scheme (CIM–PIM–PSM–Code) has been enhanced by adding a new layer “Domain Knowledge Model” for domain causal knowledge discovery (Gudas and Valatavičius, 2020). Thus, the causal modelling approach underpins the consistent EA development that ensures the validation of empirical solutions against captured causal knowledge. This presupposes the development of intelligent EA development tools that have a causal knowledge base. Model transformations and validation can be computer-aided, using causal knowledge repositories of intelligent software tools.

9. Conclusions

The study shows a way of causal modelling integration to the MODAF framework, starting from the StV viewpoint. Causal modelling is consistent with the paradigm of internal modelling, which reveals the regularity (causality) inherent in the reality domain type. The paradigm of internal modelling seeks to discover the regularities of the subject area (causal knowledge) and to create an internal model of the reality domain. The condition for causal modelling is prior knowledge of the causality of the field.

The contribution of research is bridging the gap between enterprise domain knowledge and EA framework content by the integration of meta-models as part of EA structures. Meta-models that cover not only simple process flows, but also business behaviour, i.e. causality of the activities in the domain, have been developed. In the next step, creating EA development tools, causal meta-models enables to create a layer of knowledge in the EA framework, which ensures smart EA development, allows validation of developer decisions.

The discovered causal knowledge is defined here as Domain Knowledge Model (DKM) and is tailored to the needs of enterprise application software development. The content of the DKM is the source of causal dependencies for determining the causal meta-models of the EA frameworks. The essential thing is the recognition of the defined type of causation – the circular causality, which is characteristic of the enterprise management activity. Two circular causality patterns specific to the enterprise domain – a Management Transaction (MT framework) and a more detailed Elementary Management Cycle (EMC framework) – are unified components of causal knowledge, and are used here to develop MODAF modification, that is, to create causal meta-models for StV view products StV-1, StV-2, and StV-4. The new concepts *Collapsed Capability*, *Capability Type* and *Capability Role* were introduced to specify a causality inherent to an enterprise domain.

Meta-models have enabled a causal knowledge base in the EA repository, which is a key component of intelligent EA tools. Such a smart EA development environment communicates with the EA developer; determines the compatibility of solutions with the business domain causation, validating empirical developer decisions. The case study, which is described in detail, confirms this.

Two types of validation of EA solutions are available due to causal meta-models of EA framework: the first one, assessment of the internal structure of the EA solutions against causal meta-model, and the second one, the discovery of dependencies between identified

(observation-based) capabilities. Current EA development tools visualize the created EA models, help check model syntax, but cannot perform validation of EA solution, as it has no domain knowledge models. The SAR Enterprise development example provides important technical details as causal meta-models can be integrated into a specific EA framework. The example provided shows the principled way that causal knowledge base lead to the intelligent EA development, together with the appropriate algorithms, provides a new opportunity to test and validate EA development solution, ensuring the consistency of the EA project.

The next step in the study of causal models integration with EA frameworks is based on the logical relationships of MODAF views as follows. MODAF provides a logical association of StV products and OV products, there is a direct link between the Capability and the Operational node. This led to a definition of causality-based types of Operational Nodes with different granularity: Expanded Operational Node (MT-based) and Detailed Expanded Operational Node (EMC-based).

The presented method provides an opportunity to move the EA development to smart platforms. Upgrading the core EA systems with causal models (using the MODAF example) reveals the structure of intelligent EA development software. This presupposes the development of advanced EA development tool with the ability of computer-aided validation of EA solutions in addition to expert knowledge. This would streamline the EA development process and increase the quality of software development.

References

- Anton, A. (1996). Goal-based requirements analysis. In: *Proceedings of the Second International Conference on Requirements Engineering (ICRE'96)*, 15–16 April 1996, Colorado Springs, CO, USA, pp. 136–144.
- ArchiMate (2017). *Open Group Standard. ArchiMate 3.0.1.Specification*, 2012–2017. <https://pubs.opengroup.org/architecture/archimate3-doc/>.
- Bernus, P., Noran, O. (2010). A metamodel for enterprise architecture. In: Bernus, P., Doumeingts, G., Fox, M. (Eds.), *Enterprise Architecture, Integration and Interoperability. EAI2N 2010, IFIP Advances in Information and Communication Technology*, Vol. 326. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-15509-3_6.
- Bunge, M. (2011). *Causality and Modern Science*, third revised ed. Courier Corporation, DOVER Publications, Inc., Mineola, NY
- Dardenne, A., van Lamsweerde, A., Fickas, S. (1993). Goal-directed requirements acquisition. *Science of Computer Programming*, 20(1/2), 3–50.
- Deming, W.E. (1993). *The New Economics for Industry, Government and Education*. Massachusetts Institute of Technology, Center for Advanced Engineering Study, Cambridge.
- DoD (2007). *DoD Architecture Framework Version 1.5*. Volume II: Product Descriptions. 23 April 2007.
- Dietz, J.L.G. (2006). The deep structure of business processes. *Communications of the ACM*, 49(5), 58–64.
- Emery, D., Hilliard, R. (2009). Every architecture description needs a framework: expressing architecture frameworks using ISO/IEC 42010. In: *Joint Working IEEE/IFIP Conference on Software Architecture, 2009 & European Conference on Software Architecture*, pp. 30–40. <https://doi.org/10.1109/WICSA.2009.5290789>.
- Gelžinienė, J., Gudas, S. (2015). Compatibility study of enterprise information systems modeling tools. In: *Information Technologies 2015. XX Interuniversity Conference of Master's and Doctoral Students. Vilnius University 2015. Conference Proceedings*, pp. 93–97. ISSN 2029-249X.
- Glymour, C. (2004). Critical notice. *British Journal for the Philosophy of Science*, 55, 779–790.
- Grefen, P.W.P.J. (2002). Transactional workflows or workflow transactions? In: *DEXA 2002, LNCS*, Vol. 2453. Springer-Verlag, Berlin, Heidelberg, pp. 60–69.

- Grundspenkis, J. (1998). Causal domain model driven knowledge acquisition for expert diagnosis system development. *Journal of Intelligent Manufacturing*, 9(6), 547–558.
- Gudas, S. (2012). *Foundations of the Information Systems' Engineering Theory*. Vilnius University, Vilnius (monograph, in Lithuanian).
- Gudas, S. (2016). Information systems engineering and knowledge-based enterprise modelling: towards foundations of theory. In: Kavoura, A., Sakas, D.P., Tomaras, P. (Eds.), *Springer Proceedings in Business and Economics, Strategic Innovative Marketing*. Springer, pp. 481–497.
- Gudas, S., Lopata, A. (2016). Towards internal modeling of the information systems application domain. *Informatica*, 27(1), 1–29.
- Gudas, S., Valatavičius, A. (2017). Normalization of domain modeling in enterprise software development. *Baltic Journal of Modern Computing*, 5(4), 329–350.
- Gudas, S., Valatavičius, A. (2020). Extending model-driven development process with causal modeling approach. In: Dzemlyda, G., Bernatavičienė, J., Kacprzyk, J. (Eds.), *Data Science: New Issues, Challenges and Applications, Studies in Computational Intelligence*, Vol. 869. Springer, Cham, pp. 111–143.
- Halpern, J.Y. (2015). A modification of the Halpern–Pearl definition of causality. In: *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence (IJCAI 2015)*, pp. 3022–3033.
- Hause, M., Bleakley, G., Morkevicius, A. (2016). Technology update on the Unified Architecture Framework (UAF). *INCOSE International Symposium*, 26(1), 1145–1160.
- Injun, C., Chulsoon, P., Changwoo, L. (2002). A transactional workflow model for engineering/manufacturing processes. *International Journal of Computer Integrated Manufacturing*, 15(2), 178–192.
- Kephart, J.O., Chess, D.M. (2003). The vision of autonomic computing. *Computer*, 36(1), 41–50.
- Kosanke, K. (1997). Comparison of Enterprise Modelling Methodologies. In: Goossenaerts, J., Kimura, F., Wortmann, H. (Eds.). *Information Infrastructure Systems for Manufacturing, DIISM 1996. IFIP – The International Federation for Information Processing*. Springer, Boston, MA. https://doi.org/10.1007/978-0-387-35063-9_11. Retrieved 21 October 2020.
- Lagerström, R., Saat, J., Franke, U., Aier, S., Ekstedt, M. (2009). Enterprise meta modeling methods – combining a stakeholder-oriented and a causality-based approach. In: Halpin, T. et al. (Eds.), *Enterprise, Business-Process and Information Systems Modeling. BPMDS 2009, EMMSAD 2009, Lecture Notes in Business Information Processing*, Vol. 29. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-01862-6_31.
- Medina-Mora, R., Wong, H., Flores, P. (1992). The action workflow approach to workflow management. In: *Proceedings of the 4th Conference on Computer-Supported Cooperative Work*, pp. 281–288.
- Matthew, H., Graham, B., Van Zandt Lonie (2013). *Unified Architecture Framework Profile (UAFP) Draft Table*. OMG, UPDMGroup 2013.
- Matthew, H., Bleakley, G., Morkevicius, A. (2016). Technology update on the Unified Architecture Framework (UAF). *INCOSE International Symposium*, 26, 1145–1160.
- MODAF Strategic Viewpoint (2019). *UPDM 2 Plugin 19.0 LTR documentation*. <https://docs.nomagic.com/display/UPDM2P190/MODAF+Strategic+Viewpoint>. Retrieved 21 October 2020.
- MODAF (2013). MODAF M3 version 1.2.004 2013-01-15. 152 pp. <http://en.wikipedia.org/wiki/MODAF>. Retrieved 21 October 2020.
- Morkevicius, A., Gudas, S. (2011). Enterprise knowledge-based software requirements elicitation. *Information Technology and Control*, 40(3), 181–190.
- Morkevicius, A., Bisikirskiene, L., Bleakley, G. (2017). Using a system of a systems modeling approach for developing Industrial Internet of Things applications. In: *2017 12th System of Systems Engineering Conference (SoSE)*, pp. 71–78. <https://doi.org/10.1109/SYSOSE.2017.7994942>.
- Mylopoulos, J., Chung, L., Nixon, B.A. (1992). Representing and using non-functional requirements: a process-oriented approach. *IEEE Transactions on Software Engineering*, 18(6), 483–497.
- Pearl, J. (2000). *Causality*. Cambridge University Press, Cambridge.
- Pearl, J. (2009). Causal inference in statistics: an overview. *Statistics Surveys*, 3, 96–146. <https://doi.org/10.1214/09-SS057>.
- Persse, J. (2012). *The ITIL Process Manual – Key Processes and Their Application*. Van Haren Publishing.
- Porter, M.E. (1985). *Competitive Advantage*. The Free Press, New York.
- Risk management – principles and guidelines (2009). ISO:31000:2009.
- OCEG (2009). GRC Capability model, “Red Book” 2.0. <http://www.oceg.org>.
- Rummler, G.A., Ramias, A., Rummler, R.A. (2010). *White Space Revisited: Creating Value Through Process*. Wiley, San Francisco.

- Rusinkiewicz, M., Sheth, A. (1994). Specification and execution of transactional workflows. In: Beyond, Kim, W. (Eds.), *Modern Database Systems: The Object Model, Interoperability*. ACM Press.
- Schurz, G., Gebharder, A. (2016). Causality as a theoretical concept: explanatory warrant and empirical content of the theory of causal nets. *Synthese*, 193(4), 1073–1103.
- Schekkerman, J. (2004). *Enterprise Architecture Validation-Achieving Business-Aligned and Validated Enterprise Architectures*. Institute for Enterprise Architecture Developments. http://www.enterprisearchitecture.info_27.
- Sienou, A., Lamine, E., Karduck, A.P., Pingaud, H. (2008). Towards a semi-formal modeling language supporting collaboration between risk and Process Manager. In: *Proceedings of the 2008 Second IEEE International Conference on Digital Ecosystems and Technologies*, Phitsanulok, Thailand, pp. 119–125.
- Spirtes, P., Glymour, C., Scheines, R. (2000). *Causation, Prediction, and Search*. The MIT Press. ISBN 0-262-19440-6. 543 pp.
- TOGAF Standard, Version 9.2 (2018). <https://publications.opengroup.org/standards/togaf/specifications/c182>. Retrieved 16 August 2020.
- Unified Architecture Framework (UAF) Domain Metamodel (2020). OMG Document Number: formal/19-11-05. Release Date: April 2020 Standard document URL: <https://www.omg.org/spec/UAF/1.1>. <https://www.omg.org/spec/UAF/1.1/DMM/PDF>.
- UPDM (2017). *Information technology – Object Management Group Unified Profile for DoDAF and MODAF (UPDM)*, 2.1.1. ISO/IEC 19513:2017(E). <https://www.omg.org/updm/index.htm>. Retrieved 21 October 2020.
- Von Foerster, H., Mead, M., Teuber, H.L. (1953). *Cybernetics: Circular Causal and Feedback Mechanisms in Biological and Social Systems*. Josiah Macy Jr Foundation, New York, NY.
- Zack, M.H. (1999). Developing a knowledge strategy. *California Management Review*, 41(3), 125–145.

S. Gudas is a professor at Faculty of Mathematics and Informatics of Vilnius University, and the head of Cyber-social Systems Engineering Group at the Institute of Data Science and Digital Technologies, Vilnius University. From 2005 to 2012 he was a professor at the Department of Information Systems of Kaunas University of Technology. From 2008 to 2013 he was the dean of Kaunas Faculty of Humanities of Vilnius University. His current research focuses on the causal knowledge-based software engineering. His contribution is the monograph “Foundations of the Information Systems Engineering Theory” (2012). The causal model of the enterprise domain defines the hierarchy of management transactions in the abstract space (aggregation, generalization, time), includes a systematic classification of hierarchical interactions of processes (a taxonomy of coordination). Based on causal knowledge a normalized software development life cycle is defined. In addition to the monograph, his scientific and methodological publications include 4 chapters in online books, over 180 scientific publications, 2 textbooks and 10 methodological guides for students.