INFORMATICA, 2016, Vol. 27, No. 1, 1–29

DOI: http://dx.doi.org/10.15388/Informatica.2016.74

Towards Internal Modelling of the Information Systems Application Domain

Saulius GUDAS[1,2]*, Audrius LOPATA[2]
[1]*Institute of Mathematics and Informatics, Vilnius University*
Akademijos 4, LT-2021 Vilnius, Lithuania
[2]*Kaunas Faculty of Humanities, Vilnius University*
Muitinės 8, LT-44280 Kaunas, Lithuania
e-mail: Saulius.Gudas@khf.vu.lt, Audrius.Lopata@khf.vu.lt

Received: March 2014; accepted: September 2015

Abstract. Model-driven IS engineering methods invoke the IS application domain modelling methods to acquire essential characteristics of organizational systems (enterprises). Business modelling for value creation is a relatively separate area, meanwhile it correlates with the IS application domain modelling methodologies and gives new insights for enhancement of enterprise modelling, business process modelling and BP management modelling approaches. The IS application domain and the business domain modelling are not isolated and could be investigated using the same paradigm of modelling. Yet there is some uncertainty in model-driven approaches towards the understanding of the enterprise management activities. A problematic consistency of modelling approaches indicate a need for a systemic analysis of IS application domain modelling concepts. The internal modelling paradigm is used for analysis of an enterprise management activity as a self-managed system, and hereby the lack of the conceptual basis for domain modelling in IS engineering is determined. This approach is aimed to reveal hidden information transactions of the business management activities. The understanding of the IS application domain as a self-managed system allowed to redefine such concepts as management transaction, management function and enterprise process. The metastructure of management transaction is defined and illustrated for business management layer and IS development layer.

Key words: IS application domain, enterprise modelling (EM), business process modelling (BPM), business modelling (BM), internal modelling paradigm, management transaction, management function, self-managed system.

1. Introduction

Enterprise information systems (IS) development methodologies are related with application domain modelling approaches – enterprise modelling, business process modelling, business process management modelling and business modelling. Model-driven IS development approaches are framed by MDA principles (Frankel, 2003) and promote development of model-driven engineering (MDE) methods and CASE tools. Integration of goal-driven software development methods (van Lamsweerde, 2001; Korherr, 2007) with business process modelling languages (BPMN, IDEF0, IDEF3, ARIS,

*Corresponding author.

etc.), enterprise (architecture) modelling frameworks DoDAF, MODAF, NAF, UPDM (Schekkerman, 2003), ArchiMate (Lankhorst *et al.*, 2005; Scheer and Nuttgens, 2000), as well as uprising of new Agile IS development approaches (Rosenberg *et al.*, 2005; Schwaber, 2004) are steps of advancement of the IS engineering methods and techniques.

A new complex enterprise architecture modelling methodology emerges, for instance, the Enterprise Architecture Cybernetics (Kandjani *et al.*, 2013) based on the complex systems theories. New types of software systems originate – the Complex global software systems (Kandjani *et al.*, 2012) as well as autonomous software (Kephart and Chess, 2003) require new development approaches. Currently, there are different groups of modelling frameworks and languages related to modelling of the IS domain, i.e. organizational system, namely, Business Modelling (BM), Enterprise Modelling (EM), Business Process Modelling (BPM) and Business Process Management Modelling (BPMM) (OMG, 2014). A variety of methods and aspects of organizational modelling methods (perspectives, aspects and classifications of Business Process Modelling and Information Systems Modelling Techniques) are presented in Giaglis (2001), Krogstie (2012). The traditional EM and BPM methods and variety of aspects of IS domain modelling in IS development methods have no formal proving (evidence) for consistency from this viewpoint (Giaglis, 2001; Krogstie, 2012; Lupeikiene *et al.*, 2014).

There is some uncertainty towards the understanding of the IS application domain as a complex system. The importance of modelling an organization as a complex system before eliciting its requirements in IS engineering is emphasized in Hevner *et al.* (2004), Avison and Fitzgerald (2003). A wide range of modelling aspects and consistency of modelling requires a systemic analysis of modelling concepts used for identification of essential characteristics of the application domain. The problem of an understanding of organizations as real world systems in model-driven IS development is discussed in Leonard and Ralyte (2010), Solvberg (2010), Montilva and Barrios (2004), Barios and Montilva (2004). "An understanding of the enterprise IS (EIS) application domain is a critical factor for . . .requirements engineering. . ." So, a critical factor for IS development is ". . . the domain knowledge that EIS developers are supposed to have" (Montilva and Barrios, 2004). The real world domain in the IS engineering methodologies and methods is diversely referred to as "business", "business organization", "organization" or "enterprise", "business process", etc. The systemic definition of these concepts in the context of IS engineering remains an open problem discussed as Activities World versus IS Conceptual World in Leonard and Ralyte (2010), information system and domain system in Solvberg (2010), enterprise IS (EIS) and application domain in Montilva and Barrios (2004). "A very few of them (IS developers) provide a clear and precise definition of the enterprise as a system" (Montilva and Barrios, 2004). One of the key aspects of transformations between different models is preserving semantics of the problem domain represented in different models and implemented by using different technologies. Such a situation proposes the traceability solutions as one of the ways for linking different software and system models, e.g. Pavalkis *et al.* (2013); however, such solutions do not yet cover the problem of linking the components of the enterprise as a whole.

The paper focuses on the systemic characteristics of information transactions and evaluation capabilities of the business modelling (BM) frameworks as well as the business process modelling (BPM), enterprise modelling (EM) approaches for model-driven IS development. A relevant definition of IS application domain has become a challenge for improvement of model-driven engineering (MDE) methods. The paper presents a holistic view based approach to the analysis of the relationship between IS application domain models and business domain models. The matter of investigation is closely related with the issues of business and IT alignment modelling, integration and interoperability in enterprise modelling (Chen *et al.*, 2008; Chen and Doumeingts, 2003). In business practice integrity of IT and business is an obligatory point of view (Lankhorst, 2004). For instance, the business and the IT strategic alignment model (Henderson and Venkatraman, 1990) discover the key principles of interactions between the business domain and the IT domain (i.e. the IS application domain).

So, these two domains (IS application domain and business domain) are not isolated and could be investigated using the same paradigm of modelling, i.e. the internal modelling view point to reveal the information transactions (hidden) in the business models and to use for enterprise IS development needs. This is one of the arguments why the information content of the business (management) domain is required to be investigated in a more detailed way. Note that the terms (concepts) commonly used in business models (Porter, 1985; Deming, 1993; Rummler *et al.*, 2010) to represent the essential characteristics are *primary activity, support activity, business function, business transaction, management function, organizational control, management*; however, these terms in EM and BPM methods are abstract or remain undefined. There is a distinction in the understanding of modelling concepts in business models (BM) and IS domain modelling (EM, BPM, BPMM) methods. For instance, only a few of IS domain modelling (EM, BPM, BPMM) methods include modelling constructs for the analysis of the management/control perspective or the information/knowledge/goal interactions perspective (List and Korherr, 2006). Definition of the enterprise modelling concepts "function" and "process" raise discussions, i.e. whether these two concepts denote qualitative different types of enterprise activities (Harmon, 2011; Owens, 2013; Montilva and Barrios, 2004; Scheer and Nuttgens, 2000; Barker and Longman, 1992), or they are mere elements of the same hierarchy (processes are components of functions and vice versa) (Lankhorst *et al.*, 2005; Weske, 2007). As a result, does the distinction between the content of different modelling concepts pose a real problem that, in its turn, raises difficulties for reflecting the essential characteristics of business in IS domain models?

Let us use the concept "Organizational System" as an umbrella term to denote the application domain – organizations (Ackoff, 1971). Organizations are a specific type of complex systems – organizational systems (Ackoff, 1971; Yolles, 2006). Definitions of the organizational system as a separate type of complex systems include the essential characteristics as follows: (a) organization is a purposeful system, i.e. subdivisions of an organization have a common purpose in the relation of which an organization has a functional division of labour (Ackoff, 1971): some major types of "primary activities" in Porter (1985) or typical functions of management; (b) a communication system with functionally different parts (i.e. inter-functional relations) (Ackoff, 1971) and a system-control function which operates on the basis of the feedback principle (Ackoff, 1971). Self-management is one of the qualitative characteristics of an organizational system comprising its goal-driven behaviour.

This paper is an attempt to vindicate that the external viewpoint alone is not sufficient in IS engineering, it is unsufficient to model the essential management information interactions. The presented approach is aimed to identify the informational content of the business management transactions and to define the key principles of mapping to the IS application domain modelling. This paper is organized as follows. First we will discuss the application domain modelling paradigms, organizations as a self-managed system in Section 2. Characteristics and evaluation of the enterprise domain modelling approaches is presented in Section 3. Section 4 is focused on the conceptual gap between business modelling layer and EM, BPM, BPMM approaches and incompatibility of enterprise modelling-related concepts. Internal view-based framework of management transaction is presented in Section 5. Definitions of management transaction and management function are presented and illustrated in Section 6. After that we have conclusions concerning the key elements of enterprise domain knowledge-based modelling.

2. Principles of the IS Application Domain Analysis

External modelling and internal modelling paradigms are well known principles of modelling and are explained in General Systems Theory by invoking "Black box" and "White box" concepts (Skyttner, 2002). The degree of understanding of the real world domain increases while moving from the "Black Box" model towards the "Grey Box" model and, finally, to the "White Box" model of a real world object (Skyttner, 2002). From our point of view, the obtaining of the essential information in management interactions has no systemic support in BPM languages and CASE tools, because BPM languages and BPM methodologies are built using the external modelling paradigm. In order to understand why an organization transforms information in the definite way, it is necessary to capture management information transformations as a whole – a management control system, including enterprise goals and the information feedback loop. So, there is a need to explore the internal modelling paradigm in IS application domain modelling.

2.1. *Domain Modelling Paradigms*

The qualitative differences of the external modelling and internal modelling paradigms are listed in Table 1 and are discussed in Gudas (2012a, 2012b), Gudas and Lopata (2002). The external modelling paradigm is relevant to the "black box" approach since modelling is based on the analysis of input/output of the system (i.e. an empirical modelling). Internal modelling acquires not only input/output of the system, but also requires a priori knowledge on the internal structure of system and dependencies between components (i.e. a knowledge-based modelling). Internal models are widely used in automatic control systems (Dorf and Bishop, 2011). Nowadays, internal modelling of business (organizational) systems is required for developing advanced enterprise management systems, knowledge-based IS, knowledge management systems as well as new knowledge-based IS development methods and enterprise modelling techniques.

Table 1
The qualitative differences of the external modelling and internal modelling paradigms.

Features of an external modelling paradigm	Features of an internal modelling paradigm
A black box approach – is based on the analysis of input/output of the observed system	A white box approach – is based on the modelling of a real world system as white box, a self-managed system
Analyst (an observer or expert) uses: • An acquired empirical information • Knowledge of enterprise modelling notations	Analyst (an observer or expert) uses: • An acquired empirical information • Knowledge of enterprise modelling notations • Theoretical knowledge on the nature of the problem domain: a priori knowledge of the regularities and dependencies (consistent patterns, "laws") of application domain
Emprical modelling in the context of the first-order cybernetics	Knowledge-based modelling in the context of the second-order cybernetics

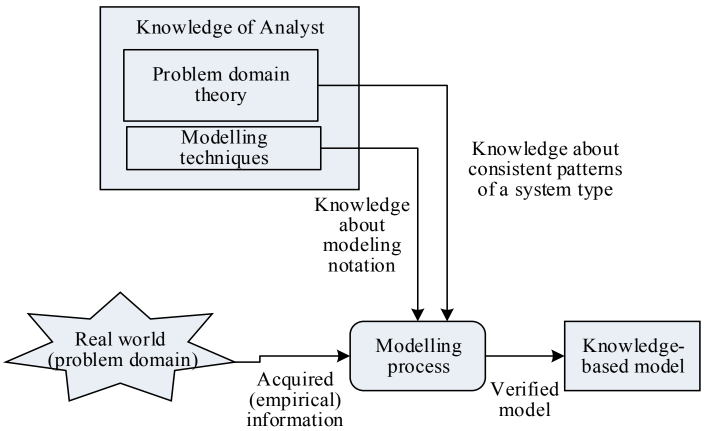


Fig. 1. Scheme of the internal modelling process.

While carrying out internal modelling (Fig. 1), the analyst (observer, expert) uses theoretical knowledge on the nature of problem domain (Gudas, 2012a, 2012b). Thus, internal modelling is knowledge-based modelling because the analyst (observer, expert) uses theoretical knowledge (methodology and methods) about consistent patterns and dependencies within the problem domain.

There are several known enterprise modelling methods that can be conditionally attributed to the internal enterprise modelling perspective, e.g. Action Workflow (Medina-Mora *et al.*, 1992), Deming's "Plan-Do-Check-Act" (PDCA) cycle (Deming, 1993), Rummler–Brache model (Rummler *et al.*, 2010). All of them are focused on the necessity for a feedback loop of enterprise management activities.

Other essential examples of internal modelling are a 'process' approach to quality management in ISO 9001:2000 developed using the "Plan-Do-Check-Act" methodology (Brown and Sondalini, 2004), the structured approach to Enterprise Risk Management (ERM) frameworks (SARM, 2010; ERM, 2011), a model of the process-based quality management system ITIL process management (ITIL, 2007), Continuous Process Improvement (CPI) life cycle in DoD (2008), Microsoft Operations Framework MOF (2014).

2.2. *Characteristics of Organizational Systems*

An organizational system is a complex phenomenon which is a hierarchical system that has several views (contexts) of analysis as follows: *organizational view* – an organization is a social/economic system. In this context, an organization consists of organizational units (departments, sub-departments) that perform goal-driven enterprise management activities; *knowledge* (*management*) *view* – an organization is a knowledge management system. This context of analysis includes goal-driven knowledge transformation activities (knowledge acquisition, storage and transformation); *information transformation view* – an organization is an information processing system. This context of analysis considers goal-driven data/information transformations; *infrastructure view* – the organization's infrastructure is concerned and includes ICT facilities and enterprise IS applications. Our approach encompasses the modelling context mentioned above, but is focused only on goal-driven data/information/knowledge interactions and transformations within the organizational hierarchy. We use the concept "management information" as a general concept to express *a data/information/message/knowledge* exchange in organizational systems.

This management information perspective is the key point in the systemic research of basic characteristics of organizational systems for IS development: "... information is the "glue" that holds an organizational structure together. Information can be used to better integrate process activities both within a process and across multiple processes" (Davenport, 1993). The essence of business "can be expressed as an exchange of messages between two or more roles" (Dubray, 2002).

Organizational Systems are a type of complex systems with a particular feature of behaviour, i.e. a self-management (Kramer and Magee, 2007; Brun *et al.*, 2009). The three-layer architecture of a self-managed systems (Goal management layer, Change layer, Component control layer) and the feedback control loop of transactions between the components of these layers refine the essential characteristics of self-management in Kramer and Magee (2007). Organizations aim to achieve a predetermined goal; information flows in organizations have the nature of closed loops (feedback), whereas organizations are defined as goal-seeking (or purposeful) systems (Johnson, 1976; Ackoff, 1971). Organizations employ feedback to achieve a predetermined goal and must have an indication of its degree of attainment available to it at all times (Johnson, 1976). Self-managed systems, self-organizing and self-adaptive systems are types of complex systems with a similar architecture which comprises a feedback control loop (Kramer and Magee, 2007). The important feature of these systems is that "all these systems use internal representations of global properties or goals" (Brun *et al.*, 2009). *A managed element* is an obligatory element of a self-managed system, i.e. *a managed element* "produces" the major output of a system; it is an object of monitoring and control within the feedback *control loop*.

Theories of complex systems are applied in enterprise architecture modelling, i.e. Enterprise Architecture Cybernetics (Kandjani *et al.*, 2013) emerges, and they are applied

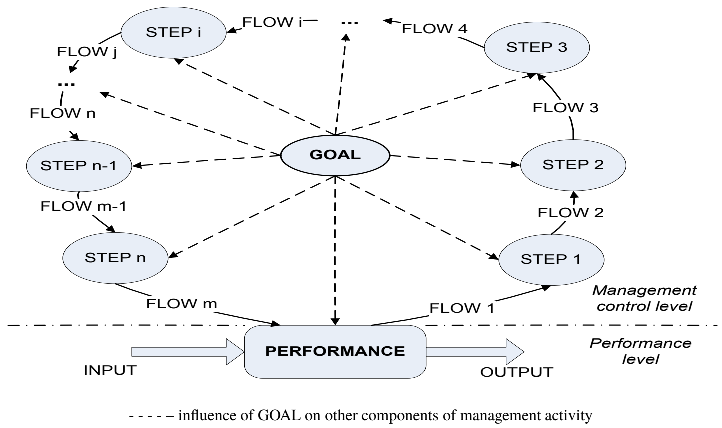


Fig. 2. Internal perspective of management activity (a framework of a self-managed system).

in autonomous software systems development as well Kephart and Chess (2003). The complexity of Global software development project management requires focusing on the theories of self-adaptive systems (Kandjani *et al.*, 2012). An example of a self-adaptive system in software development is an "autonomic element" introduced in Kephart and Chess (2003). The structure of "autonomic element" includes a "managed element" and "autonomic manager", which comprises the following steps of the knowledge-driven control cycle: monitor, analyze, plan, execute.

2.3. *The Framework of Management Activity*

Management is a process "consisting of planning, organizing, actuating and controlling, performed to determine and accomplish the objectives by the use of people and resources" (Tripathi and Reddy, 2008). The framework of management activity is summarized in Fig. 2 as a management cycle C (Goal, Performance):

$$\begin{aligned} &\text{C (Goal, Performance)} \\ &\quad = \{\text{STEP 1 (Goal), STEP 2 (Goal)}, \ldots, \text{STEP } i \text{ (Goal)}, \ldots, \text{STEP } n \text{ (Goal)}\}. \end{aligned}$$

There is a qualitative distinction between the understanding of management activity as a goal-driven system and as a self-managed system. *The external modelling perspective*: in the *framework of a goal-driven system* the component GOAL is external to the management cycle; thus, the interactions between STEPS (i.e. primary functions of management) through FLOWS (management information) are directed by the external component GOAL from the higher level. *The internal modelling perspective*: in the *self-managed system* (Fig. 2) all internal components of the management cycle (FLOWS, STEPS, and the component GOAL) are integral parts of the control cycle, all elements are on the same level.

The feature "self-managed" stands for the ability of a system's goals to influence the content of interactions within the management cycle, as well as the ability of a system to formulate and change goals. Thus, the IS application domain model includes the following element types to conform to the essential properties of a self-managed system (Fig. 2): GOAL (objective, requirement, capability, constraint), STEP (function, process, activity, action, task), FLOW (data, information, knowledge, message) and PERFORMANCE (material/energy transformation, production, etc.), management information flows between STEPS; management information interactions between GOAL and STEPS and between GOAL and FLOWS; a topology of management activity is a cycle (or control loop), i.e. typical elements of a self-managed system. These essential properties of management activity are used as criteria for the analysis of IS domain modelling methods.

3. Evaluation of the IS Application Domain Modelling Approaches

Frameworks and languages for IS application domain modelling include perspectives of modelling classified as Enterprise Modelling, BP modelling, BP management modelling approaches. The particularity of the accomplished analysis is that the IS domain is seen as a type of complex systems, i.e. an organizational system. Such systems comprise self-management as the essential feature of behaviour.

3.1. *Characteristics of Enterprise Modelling Frameworks*

The term enterprise modelling (EM) is a collective name for the use of models in enterprise engineering and enterprise operation (Bernus, 2001), enterprise architecture modelling: organizational architecture, business architecture, application architecture, technical infrastructure (Schekkerman, 2003). Enterprise modelling frameworks are used for the development of business process re-engineering methods as well as for architecture-driven IS engineering. The most popular EA frameworks aim at enterprise architecture modelling: TOVE, PERA, CIM-OSA, GRAI-GIM, GERAM (1999), DODAF, MODAF, NAF frameworks, ArchiMate, UPDM languages for enterprise architecture modelling (Schekkerman, 2003; Morkevicius *et al.*, 2013).

Enterprise architecture models mainly consider the high level view of operational interactions between enterprise processes and activities, organizational hierarchy, organizational units, system level interactions of services, information flows, enterprise strategies and goals (Wortmann *et al.*, 2001; Morkevicius *et al.*, 2013). For example, the perspectives (views, viewpoints) in the MODAF and NAF frameworks and UPDM are as follows: strategic view, operational view, system view, acquisition view, service view, technical view, custom viewpoint (UPDM). The ArchiMate language identifies the Business, Application and Technology layers, and uses the structural (static) aspect, behavioural (dynamic) aspect, internal view, external view, individual aspect, and collective aspect of modelling (Lankhorst *et al.*, 2005). Business strategy and IT strategy (capabilities) alignment framework perspectives (views) are as follows: business strategy, IT strategy, business infrastructure, and IT infrastructure (Henderson and Venkatraman, 1990). However, only a few EA frameworks have concepts for decision making modelling

(GRAI-GIM, GERAM), i.e. only a few of EA frameworks deals with enterprise management/control perspective (Whitman *et al.*, 2001). The EM methodologies (MODAF, ArchiMate, UPDM) include an addition of the following concepts: vision, strategy, capability, business process, business service, application service (Lankhorst *et al.*, 2005; UPDM, 2013).

The referred features indicate that EA frameworks are the consequence of the external paradigm. Thus, EA frameworks are mainly concerned with the high level view of interactions between enterprise processes and activities, information flows, organizational units, enterprise strategies and goals (Wortmann *et al.*, 2001; Morkevicius *et al.*, 2013).

3.2. *Characteristics of Business Process Modelling Approaches*

Business process modelling (BPM) is a set of methods and tools used to describe the current or future state of a business process. Business process modelling comprises modelling techniques to drive the re-structuring and software development process. BP models mainly deals with enterprise processes (functions) and activities, information and material flows, events, organizational units and external sources of flows.

The Object Management Group (OMG) has prepared a series of documents for the unification of the BPM terminology and tools: MDA approach, Business Process Model and Notation (BPMN, 2011), Semantics of Business Vocabulary and Business Rules (SBVR, 2010), Business Process Definition MetaModel (BPDM, 2008), Business Motivation Model (BMM, 2008), Unified Modelling Language (UML), Organization Structure Metamodel (OSM, 2009). The key concepts of the Model Driven Architecture (MDA) approach define the development of software systems as a sequence of transformations starting from mapping the Computation Independent Model (CIM) (CIM is sometimes called a domain model) to the Platform Independent Model (PIM). The major concepts developed in BPM methods (languages and techniques) are as follows: object, entity, user, agent, process, activity, function, state, event, transition, message, flow (information flow, control flow, material flow), association, aggregation, generalization (inheritance) (BPMN, 2011; UML, 2014; Weske, 2007; ODM, 2009). An attempt for extending the BPMN business process model with the SBVR business vocabulary and rules is presented in Skersys *et al.* (2012).

Modelling perspectives, aspects and views in BPM languages (PSL, DFD, IDEF0, IDEF3, BPMN, ARIS, KAOS, F3, etc.) are as follows: *information view, organization view, decision view, function view, dynamic view, economic view* in Giaglis (2001); perspectives of model-based development (data perspective/information perspective/structural perspective; process perspective/functional perspective; event/behaviour perspective), perspectives of conceptual modelling (organizational perspective; goal and rule perspective; object perspective; communication perspective; actor and role perspective; topological perspective), role perspective/roles of actors performing processes presented in Krogstie (2012), and views (function view, control view (workflow view), data view, organizational view, product/service view, resource view) in ARIS.

The essential characteristics of business process modelling approaches are reviewed in Table 2. The external modelling paradigm is a methodological background applied to

Table 2
Characteristics of the BM, BPM and BPMM approaches.

Components	Business modelling (BM) approaches		BPM and BPMM approaches
	PDCA cycle (E. Deming)	The Value Chain Model (M. Porter)	BPMN, IDEF3, ARIS, BPMM, etc.
Modelling paradigm	Internal modelling	Internal modelling	External modelling
GOAL	GOAL is not defined explicitly but is declared in context	GOAL is not defined explicitly, but is considered in context	GOAL as an element of the model is not defined
STEPS:			
Types of STEPS	Types of STEPS: Step 1 – Plan, Step 2 – Do, Step 3 – Check, Step 4 – Act	Types of STEPS: Primary Activities (PA); Support Activities (SA)	Elements denoting a flow transformation (Process, Activity) are not classified by TYPE
The sequence of STEPS	The sequence of STEPS is defined as the cycle: PDCA	The sequence of the PA is defined. The sequence of the SA is not defined	Not defined
Interactions between STEPS	Interactions between STEPS are defined by PDCA cycle	The interactions between PA and SA are implicit, but not identified	Not defined
FLOWS: Types of Flows	Types of Flows not defined explicitly	Types of FLOWS are not defined explicitly	Types of FLOWS: data, information, message, knowledge, material flow
Interactions:			
Between STEPS	Types of interactions in the PDCA are not defined explicitly	Types of interactions PA/SA are implicit, but not defined	Not defined
Between GOAL and STEPS	Types of interactions GOAL/STEP are not defined	Types of interactions GOAL/STEP are not defined, but implicit	Not defined
Between GOAL and FLOWS	Types of GOAL/FLOW interactions are not defined	Types of GOAL/FLOW interactions are not defined	Not defined
A feedback (control) loop	Topology of the PDCA model is a cycle, i.e. a control loop is defined	The feedback loop is not defined explicitly	Feedback control loop is not defined as essential constraint or requirement
Material transformation	Semantics of the STEP "Do" includes the meaning material transformation	Semantics of PA include material transformation	Material transformation is not defined as a Type
Self-management feature	The PDCA is a self-managed system: it presumes goal-driven transactions and self-control of STEPS	The VCM is a self-managed system: it includes a goal-driven feedback interaction between PA and SA	Self-management feature is not defined or identified

BPM. As a consequence, some essential internal dependencies of the problem domain are missed in the BPM methods. Namely, the management/control dependencies (constraints) and goal-driven data/information/knowledge transactions have a slight conceptual representation in BPM (List and Korherr, 2006).

3.3. *Characteristics of Business Process Management Modelling Methods*

The Business Process Management Modelling (BPMM) deals with the analysis and management of operational business processes and "can be considered as an extension of classical Workflow Management . . . systems and approaches" (van der Aalst *et al.*, 2003). The business process management modelling (BPMM) is an approach to making a set of more effective and efficient activities of organizations. BPMM is closely related with Workflow Management and such approaches as BPA (business process analysis), BAM (Business Activities Monitoring) (van der Aalst *et al.*, 2003) and Enterprise IT management approaches such as presented in Henderson and Venkatraman (1990).

Specifications concerning certain aspects of business process management are developed by OMG: BMM – Business Motivation Model (BMM, 2008), BPMM – Business Process Maturity Model (BPMM, 2008) based on the five-level improvement model for software development management described in Humphrey (1993).

The characteristics of BMM and BPMM are introduced in Table 2. Equally, we might say that business models (BM) such as Porter's Value Chain Model (Porter, 1985), Shewhart's cycle, Deming's PDCA cycle (Deming, 1993), Rummler's approach (Rummler *et al.*, 2010) are to some extent business (process) management models at a high-level of abstraction.

3.4. *Characteristics of Business Modelling Frameworks*

Business models focus on value creation, and goal-driven governing of business processes for value creation. Business model frameworks included in the survey are examples of goal-driven management of business organizations. The importance of feedback (reciprocity, reflexivity) in business enterprises is a recognized feature in business modelling approaches, that is, the essential condition for self-management feature of organizational systems. "A business model is a representation that captures the structure and dynamics of the target organization in which the EIS is to be deployed" (Montilva and Barrios, 2004); business models essentially focus on value creation and customers (Osterwalder *et al.*, 2005).

We will focus on the following internal characteristics of recognized business models: Porter's Value Chain Model (Porter, 1985), Deming's PDCA cycle (Deming, 1993; Moen and Clifford, 2014), Rummler's approach (Rummler *et al.*, 2010). The Shewhart cycle has been used to describe the idea of production viewed as a system from the standpoint of quality control (it was first introduced in 1939) (Moen and Clifford, 2014). Deming's PDCA cycle (Plan, Do, Check, Act) is well defined as an iterative four-step management method used in business for the control and continuous improvement of processes and products (Moen and Clifford, 2014; Deming, 1993).

A systemic approach to business modelling research, i.e. "business modelling as a systemic instrument", is presented in Cocchi (2012). The classification of business models gives four different perspectives of BM (essentialist, pragmatic, functionalist, systemic). An important conclusion about the trends of BM methodologies in Cocchi (2012) states

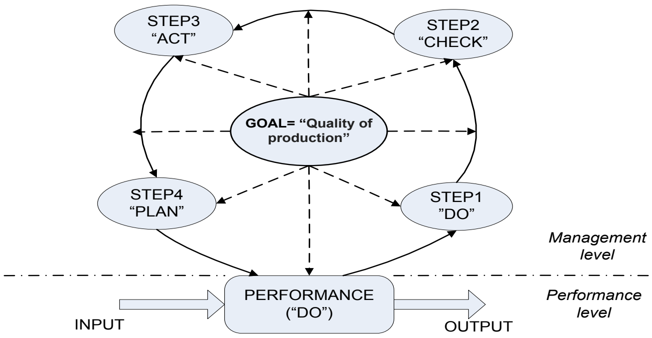


Fig. 3. The modified PDCA cycle corresponds to the framework of a self-managed system (Fig. 2), yet with some points of uncertainty.

that “BM studies progress from a close to an open perspective, in which BM knowledge is contested between different branches of science”, i.e. between BM community (business analysts) and EA/BPM community (and IS developers).

Some researches focus on the development of business modelling approaches for IS development, i.e. Business Action Theory (BAT), Dynamic Essential Modelling of Organizations (DEMO), the Resource–Event–Agent framework (REA), UMM methodology (Bergholtz *et al.*, 2003), ontology-based business modelling (Andersson *et al.*, 2005, 2006); a unified framework for business models and process models in e-commerce developments is presented in Bergholtz *et al.* (2003). The idea of aligning business and business processes formulated in Andersson *et al.* (2005) stands for employing business models as the foundation and “to base process models on business models”. Some architectural structures of business and IT integration based on the modifications in Henderson and Venkatraman (1990) framework are presented in Gudas (2012b).

The PDCA cycle is one of the best examples of a business model which corresponds to the framework of a self-managed system (Fig. 2). Let us consider W.E. Deming’s PDCA cycle (Plan, Do, Check, Act) as a (standard) business model, i.e. a base for internal analysis and comparison of other popular business models. The PDCA cycle business activities are defined as follows (Fig. 3): Step 1 – “Do (Do, Observe and collect data)”, Step 2 – “Check (Did things happen according to plan?)”, Step 3 – “Act (How to improve next time?)”, Step 4 – “Plan (What resources, criteria, limits?)”. All the steps of the PDCA cycle are managed by means of business Goal “Quality of production” (Quality of processes and products). Component “Performance” (i.e. production) corresponds to the content of PDCA step “DO”.

The PDCA cycle presumes a system of business management interactions, i.e. a system of goal-driven transactions that aim at controlling the PDCA steps and focus on the goal “Quality of production”. This is the model of a complex business system with the self-managing feature. So, Deming’s PDCA cycle (Fig. 3) corresponds to the frame of a self-managed system (Fig. 2), however some components are not explicitly defined, namely the types (identity) of the FLOWS between the STEPS, as well as information interactions between STEPS in the control loop, between GOAL and STEPS, and between GOAL and FLOWS.

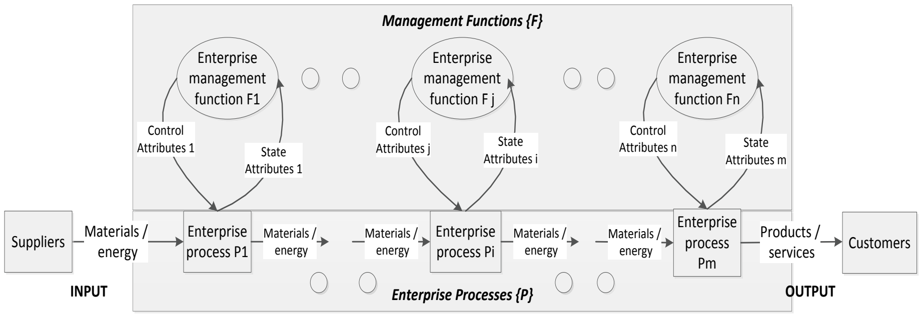


Fig. 4. The Detailed VCM refines the content of a feedback loop (control cycle) between management functions F and enterprise processes P.

Another business model – Porter's Value Chain Model (VCM) – defines the top-level structure of a business enterprise; however, it refines the principal components of business management (Porter, 1985). The VCM identifies two types of activities, i.e. a sub-set of Primary Activities and Support Activities. The content of interactions between these two sub-sets of activities is implicit and not specified in the Value Chain Model in Porter (1985). The essential characteristics of M. Porter's Value Chain Model from the internal modelling viewpoint are summarized in Table 2.

The Detailed VCM (DVCM) – a structural decomposition of the VCM was developed and is presented in Gudas and Lopata (2002, 2009). The DVCM aims at refining information interactions between the VCM components, defined in Table 2: support activities are information transformation activities by definition of the VCM, and are classified as enterprise management functions (F); primary activities are material flow transformation activities by definition, and are classified as enterprise processes (P). Enterprise management functions direct enterprise processes towards a pre-defined business goal; thus, interactions between these two sub-sets (F and P) have to be two- directional goal-driven transference of information, as a result of which a goal-driven feedback control cycle is defined (Fig. 4).

The Detailed VCM (DVCM) specifies content of interactions ($F \times P$) between *enterprise management functions* (F) and *enterprise processes* (P). Interaction ($F \times P$) are supposed to be a information feedback loop directing an organization towards the desired goal. Information flows state attributes of enterprise processes (P) are transferred to management functions (F), and feedback information flow "Decisions (control attributes)" directs enterprise processes (P). Therefore, this control view-based decomposition of internal interactions within the Detailed VCM (DVCM) corresponds to the architecture of a self-managed system (Fig. 2). Essentially Porter's Value Chain Model is a generalized representation of business enterprise with tacit features of a self-managed system.

The Rummler–Brache model of enterprise for performance improvement is constructed as an adaptive system: "Every organization must be an adaptive system operating as a part of a super-system" (Rummler *et al.*, 2010). Organizations are processing systems, hierarchical value creation systems with a feedback loop between three levels of management: Enterprise (Management system), Business (Management system) and Performance level. “Organization has to PLAN, DESIGN and MANAGE performance at three levels: organization, process and job” (Rummler *et al.*, 2010). Also, feedback loops are identified at every level of management between the components of a model. The important proposition about the Rummler–Brache model introduced in Harmon (2010) states that “each process or activity must be managed …”, i.e. in this way a goal-driven behaviour and a self-management feature is clarified. Thus, the Rummler–Brache model of an enterprise comprises the goal-driven behaviour (“for performance improvement”) of an organizational system, which essentially is the feature of self-management, though not abstracted as a formal business management framework. The following are some other business modelling frameworks comparable with the PDCA cycle (Fig. 3), and, consequently, with the meta-model of a self-managed system (Fig. 2): The Action Workflow model is a cycle (loop) which consists of unidirected actions (Proposal, Agreement, Performance, Satisfaction) aimed to fulfill the predefined goal “Satisfaction of customer” (Medina-Mora *et al.*, 1992); The Management Process model in Tripathi and Reddy (2008).

This analysis shows that business models for the original purpose – to manage organizational units and monitor internal activities and interactions – is by nature a result of internal modelling. Business modelling frameworks (e.g. Deming PDCA cycle, Rummler–Brache model, etc.) correspond to the internal modelling paradigm to a higher degree than the enterprise modelling, business process modelling, and business process management modelling methods used in IS engineering. On the other hand, business models are on high level of abstraction, and structural decomposition is required to determine implicit components and information interactions. Hence, the internal modelling paradigm is a rational way to align the business models with other modelling approaches, namely, the enterprise models and business process models.

4. On the Incompatibility of Enterprise Modelling-Related Concepts

The MDA methodology defines coherent transitions between modelling layers (CIM, PIM, PSL), i.e. MDA is based on the integration (mapping) of different modelling layers. The IS domain analysis based on the framework of a self-managed system (presented in Fig. 2) determines the differences of the modelling concepts used in these approaches. The incompatibility of modelling-related concepts evokes difficulties for the mapping between modelling layers and views, i.e. it creates a conceptual gap between BM and EM, BPM, BPMM approaches. The demand and certain ideas for the integration of BM, EM and BPM can be seen in the modelling approaches (Andersson *et al.*, 2005; van der Aalst *et al.*, 2003; Turban *et al.*, 2010; van der Aalst, 2013; Cocchi, 2012), namely: “Business Modelling is not Process Modelling” (Gordijn *et al.*, 2000), “The root cause is that a business model is not about process but about value exchanged between actors. Failure to make this separation of concerns leads to poor business decision-making and inadequate business requirements” (Gordijn and Akkermans, 2003); Basing process models on business models (Andersson *et al.*, 2005); “BM studies progress from a close to an open perspective, in which BM knowledge … is contested between different branches of science…" (Cocchi, 2012); "Business process models should have a formal foundation…" (van der Aalst *et al.*, 2003).

Bridging of business modelling experience (BM frameworks) and the modelling techniques in IS development (EM, BPM, BPMM) requires a new understanding of concepts (in the context of Organizational Systems) such as: function, process, management function, management feedback loop (control cycle), and management transaction. The definitions of business modelling-related concepts as well as those in the EM and BPM approaches are fuzzy; modelling-related concepts are defined universally, the definitions are incomplete for the internal modelling of organizational systems and the IS application domain (see Fig. 1). Business process modelling and enterprise modelling constructs are, firstly, primitive elements, i.e. they are defined as elements (have no structure), and, secondly, there are only a few concepts in EM or BPM notations with Types defined in language specifications (notations). For example, only concept Event in the BPMN has Types of Event specified in the notation of language (BPMN, 2011). Meanwhile, an inherent characteristic of business models is complexity: business models are frameworks with pre-defined structure and defined semantics of its elements, for instance, the PDCA cycle, Value Chain Model, etc. Thus, building blocks of business models (BM) are considered as complex units (white boxes).

The distinction between the content of the concepts "process" (*primary activity*) and "function" (*support activity*) is a key factor in the business modelling frameworks, and this difference should be mapped for the modelling methods in IS development. This is emphasized by IS methodology researches and in some enterprise frameworks (Owens, 2013; Lankhorst *et al.*, 2005; Montilva and Barrios, 2004; Scheer and Nuttgens, 2000; Barker and Longman, 1992; DoD, 1994). Definitions of major concepts relevant to the subject of IS domain modelling are, for instance, in TOGAF (2009), ITIL (2007). The definition of business process in the ITIL glossary states that "the business process contributes to the delivery of a product or Service to Business Customer". It is focused on the material result, i.e. on the material transformations. Meanwhile, the definition of a *function* is fuzzy: "A team or group of people and the tools they use to carry out one or more processes or Activities" (ITIL, 2007) and is somewhat similar to the definition of "business process". In TOGAF, the concept *Function* is used to describe a unit of business at all levels of granularity comprising value chain, process area, capability, business function, and function as an elementary unit of business (TOGAF, 2009). The definitions of the concepts *Activity, Function, Objective, Process Control* in ITIL glossary (ITIL, 2007) support the usage of similar concepts in internal modelling of the IS application domain. An important issue relates to ITIL (2007) definitions of such concepts as *Transaction, Management information and Data-to-Information-to-Knowledge-to-Wisdom* (DIKW), which is the basis of defining similar concepts in the internal model of the IS application domain.

In the most BPM and EM approaches the process perspective is parallel to the functional perspective (Krogstie, 2012), and the qualitative differences of these two perspectives are not identified or discussed. A detailed analysis of the content and usage of the concepts *enterprise process, enterprise function* (management function) and the interaction between these key terms from internal enterprise management modelling perspective

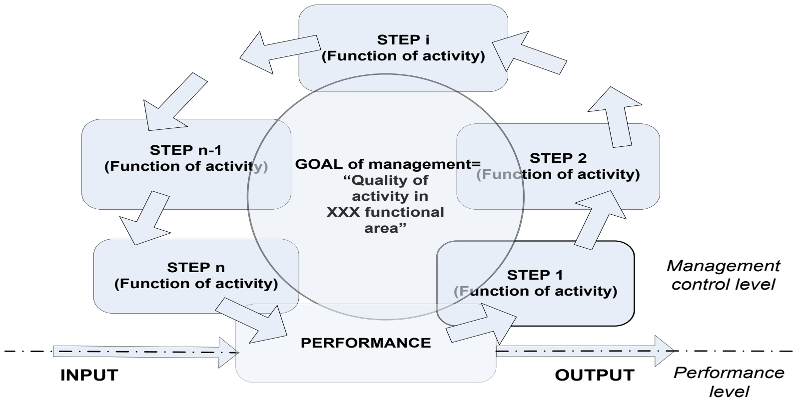


Fig. 5. Architecture of the management activity: management is a self-managed system.

is described in Gudas (2012a). One of the alternatives for IS domain modelling is enterprise management modelling approach described in Gudas (1991, 2012a), Gudas *et al.* (2005), Lopata and Gudas (2009). Management information interactions are identified by means of the Detailed Value Chain Model (Gudas and Lopata, 2009) and decomposed using the elementary management cycle (EMC) – modelling concept with formally predefined complex structure.

5. Internal Modelling of Management Activity

Management activity is a real world phenomenon and an inherent characteristic of an organizational system. There are many definitions of management in literature, most of whom are universal, for example, a *management* is defined as a process "consisting of planning, organizing, actuating and controlling, performed to determine and accomplish the objectives by the use of people and resources" (Tripathi and Reddy, 2008).

A management activity is a set of "primary functions", for example, *Forecasting, Planning, Organizing, Commanding, Coordinating, Controlling* as defined in Wren *et al.* (2002) or – by other scholars – *Planning, Operating, Directing, Controlling* (Tripathi and Reddy, 2008). Thus, a management activity, first of all, includes a predefined sequence of steps, i.e. "functions of management" (for instance, planning, operating, directing, controlling, etc.), and, second of all, a management activity is an iterative process (a cycle of management steps) (Tripathi and Reddy, 2008). The goals of "management" (business goals) are an obligatory component of a management activity that is declared implicitly, yet every function of management is a goal-driven activity.

Management in business and organizations consists of the organization and coordination of the activities to achieve the goals and objectives of a business organization. Management is a self-managed (goal-driven) system and is iterative in nature (Fig. 5). A management activity consists of primary functions (a STEP is a function of activity), whereas the architecture of a management activity includes the following two levels: management control level and performance level. The management control level is a sequence of definite STEPS (primary functions) composing a feedback control loop with the performance level activities.

It is important to note that the topology of the management activity (framework in Fig. 5) is analogous to the topology of the autonomic computing element (Kephart and Chess, 2003). The similarity is explained as the result of modelling the same type of complex systems, i.e. systems that have the feature of self-management (self-adaptiveness): both frameworks consist of a managed element and an autonomic manager, the sequence of steps and composition of a feedback control loop; the internal behaviour is driven by goals that are embedded in Kephart and Chess (2003). However, the domain of modelling is different (see Table 2): the autonomic element (Kephart and Chess, 2003) is a model of a self-adaptive software component (IT/IS domain), while the management activity framework (Fig. 5) is a self-management component within the Enterprise domain. This compatibility between the topologies of these two frameworks is a good example of mapping between different modelling stages (layers) in enterprise IS engineering: business domain modelling (enterprise modelling/business process modelling) and software domain modelling (software design).

The decomposition of the management activity (as a self-managed system) in Fig. 2 reveals the feedback control loop of STEPS and FLOWS as the essential condition in domain modelling. The essential and obligatory feature of the management activity is the influence of GOAL on STEPS and FLOWS. Two levels of the management activity are refined in Fig. 2: management control level and performance level. Furthermore, the model of the management activity (Fig. 2) illustrates the difference between the concepts "enterprise management function" and "enterprise process" from the internal modelling perspective, namely:

1. *An enterprise management function* is a part of the management activity (items at management control level in Fig. 2) which consists of goal-driven STEPS (functions of activity) and which aims at controlling the performance (*an enterprise process*);
2. *An enterprise process* is a part of the management activity (the item PERFORMANCE at performance level in Fig. 2) which comprises goal-driven material/energy transformations that form a system's output.

The topology of the management activity model in Fig. 2 is a cycle because the feedback loop is required to ensure controlling of performance. Thus, from this structural point of view, a "management activity" is an aggregate of a definite set of required elements: functions (Steps), a process (Performance) and an obligatory requirement for feedback loop. The feedback control loop is the obligatory feature of management transaction, and is indicated in the structure of the management activity as element "Feedback" (Fig. 6).

The example of the management activity according to Wren *et al.* (2002) is depicted in Fig. 7. The functions of management activity (Steps) are "primary functions", i.e. *Commanding, Forecasting, Planning, Organizing, Coordinating, and Controlling*. The term "primary function" herein means an abstract type of a required activity (carried out by organizational units); it is not directly related with a definite sub-domain (a functional area,

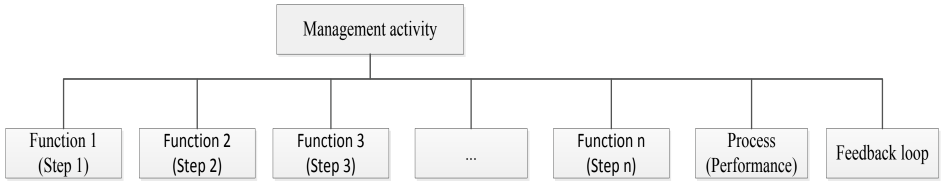


Fig. 6. The management activity is an aggregate.

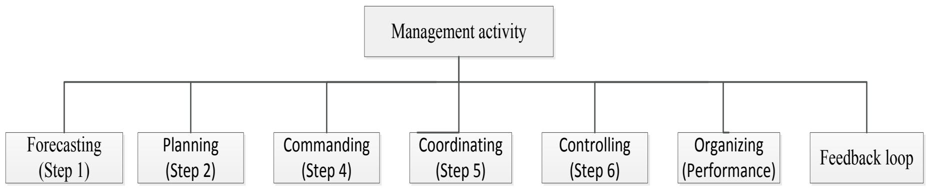


Fig. 7. The structure of a management activity according to Fayol.

e.g. finances, materials, logistics, production, services, etc.), but is a required activity in governing all sub-domains.

6. Definitions of Management Transaction and Management Function

The concepts *management transaction* and *management function* are examined as management information transformations dissociated from organizational units, resources or other infrastructural components. The internal modelling approach towards the understanding of concepts "management", "management transaction", "function of activity" and "management function" aims at identifying the major components for IT-based management modelling. The definition of *management function* from the internal viewpoint is based on the understanding of management activity as a self-managed system (Fig. 2) and closely related with the definition of *management transaction* (Fig. 8).

A management transaction is a cycle of goal-driven transformations of management information between an enterprise management function F and an enterprise process P (Level 1 in Fig. 8); here information flow S denotes state attributes of the enterprise process P, information flow D denotes controls (decisions), and Goal (G) is the context of $(F \times P)$ interactions – all components and interactions of management transaction are goal-driven.

The concept "a management information" is an abstraction, which consists of the (data, information, goals, rules, directives, constraints) used to define a content of FLOWS. Management information (in Fig. 8) also is a type of relationship between GOAL and STEPS, and between GOAL and FLOWS. Data/information usually defines some (a) measured, (b) interpreted, (c) calculated data; goal defines orchestration rules for the enterprise functional area management and control; knowledge defines models, rules and directives used for information interactions orchestration in the some functional area focused for manage-

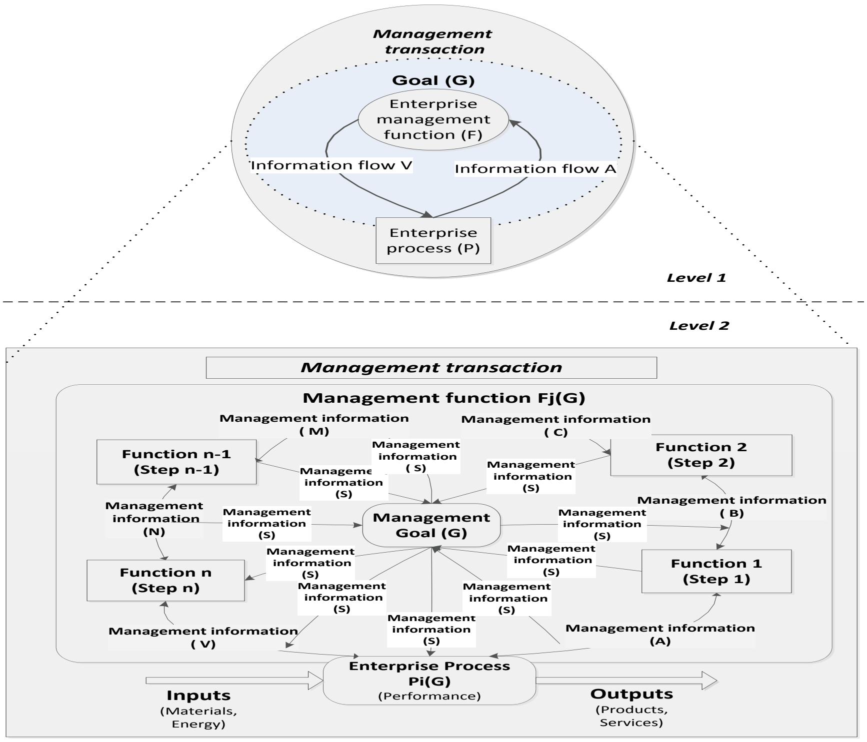


Fig. 8. Framework of a management transaction.

ment and control. All these parts (in Fig. 8) compose a complex component of management transaction in definite functional area, defined here as the elementary management cycle (EMC). Management information (S) between GOAL and STEPS (Functions) defines rules and directives for identification (or modification) of the content of the STEPS (Functions), i.e. the logic of information transformations in STEPS is goal-dependent. Management information (S) between GOAL and FLOWS defines rules and directives for identification (or modification) of a set of attributes of FLOWS (i.e. collection of attributes of FLOWS is goal-dependent).

The framework of a management transaction determine the complex structure of abstract management function $F_j(G)$ and interaction with abstract enterprise process $P_i(G)$ through management information flows (Gudas, 2012a; Gudas *et al.*, 2005). The semantics of the FLOWS "management information (A), management information (B), ..., management information (V)" between STEPS in the framework of management transaction (Fig. 8) is different and depends on the particular model of the managememnt transaction.

Considering the composition of a management transaction (Fig. 8), every *management function* is defined as an assembly of the primary functions required to control some process in the sub-domain (a functional area of enterprise).

- **Definition of a management transaction for business modelling**

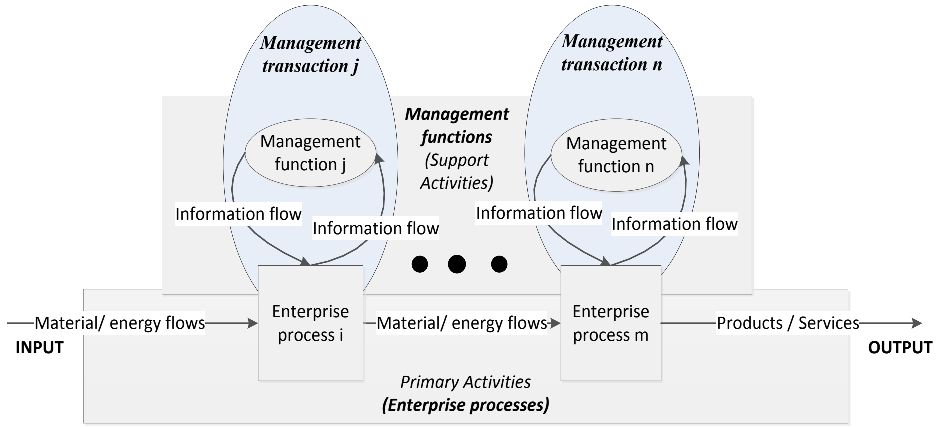


Fig. 9. The Detailed Value Chain Model as a system of management transactions.

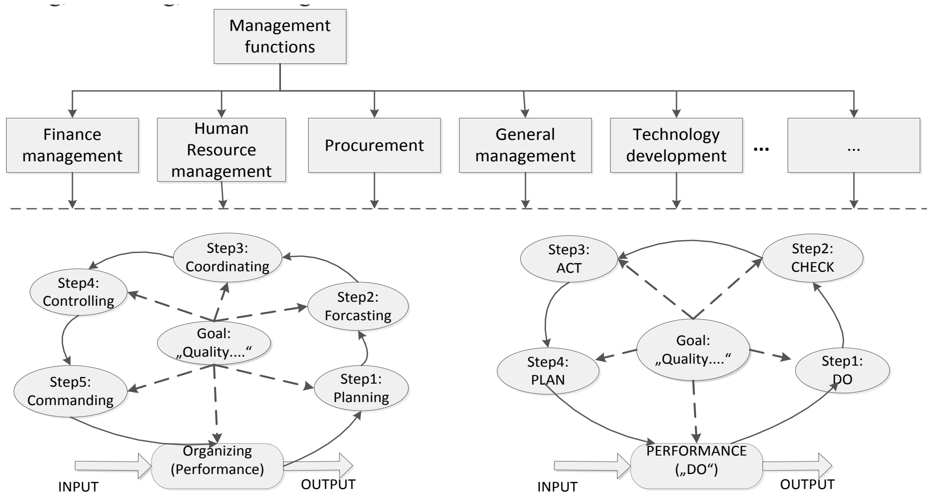
The required enterprise management functions generate management transactions in the sub-domains (functional areas) of enterprise. For instance, the support activities of the Value Chain Model (Porter, 1985) correspond to the *types of functional areas* of enterprise: finance management, human resource management, procurement, general management, technology development – these are examples of the sub-domains of enterprise management. Thus, from this point of view the support activities of the Porter's Value Chain Model are "management functions" classified by the enterprise sub-domains, i.e. finance management function, human resource management function, etc.

The Detailed Value Chain Model represented in Fig. 9 was developed as an enterprise management model for needs of IS engineering (a modified version in Gudas and Lopata, 2009). The Detailed Value Chain Model is defined as a system of management transactions in the sub-domains (functional areas) of enterprise.

Two examples of deep structure of the management transactions as assemblies of the internal steps ares presented in Fig. 10: according to Wren *et al.* (2002) in Fig. 10(a) and according to Deming's PDCA cycle in Fig. 10(b). The sub-domains in Fig. 10 are identified according to Porter's Value Chain Model. According to Fayol, the obligatory set of management activities (in any functional area) is as follows: Forecasting, Planning, Organizing, Commanding, Coordinating, Controlling. Note: however, according to Tripathi and Reddy (2008), this set is different, i.e. Planning, Operating, Directing, Controlling.

- **Definition of a management transaction for enterprise IS engineering**

A detailed composition of the management transaction (see Fig. 8) for needs of IS engineering is defined as the elementary management cycle (EMC) (Gudas, 1991, 2012a; Gudas *et al.*, 2005) and is depicted in Figs. 11 and 12. The elementary management cycle (EMC) is considered as an essential (unified) building block of an enterprise management system, i.e. a standard self-managed component of the enterprise (Gudas, 2012a). Four types of management steps are identified in the EMC: IN – interpretation (data acquisition), DP – data processing, DM – decision making, RE – realization of decisions. The elementary management cycle (EMC) includes an enterprise management Goal (G),



(a) (b)

Fig. 10. Decomposition of management functions considering: (a) the management steps according to Fayol; (b) Deming's PDCA cycle.

goal-driven procedural components (the management cycle steps IN, DP, DM , RE) and goal-driven management information flows (A, B, C, D, V) (Gudas, 2012a):

$$\mathrm{EMC}(F_j, P_i) = \big(P_i(A, G) \to \mathrm{IN}(A, B, G) \to \mathrm{DP}(B, C, G) \to \mathrm{DM}(C, D, G) \\ \to \mathrm{RE}(D, V, G) \to P_i(V, G)\big), \qquad (1)$$

here: A – a flow of state attributes of process P_i, i.e. raw data set, which is required in terms of goal G; B – a flow of systematized (interpreted) raw data required in terms of goal G; C – a flow of processed data, output of the Data Processing (DP) step, corresponds to goal G; D – a flow of management solutions, output of the Decision Making procedure (DM), corresponds to goal G; V – a flow of controls for process P_j, output of the Realization step (RE), corresponds to enterprise goal G.

Important is the association between the general level framework of the management transaction (Fig. 8) and partial level conceptual model of IS application domain in Fig. 11 – an elementary management cycle (EMC). For instance a flow *management information* (A) is "process state attributes" in Fig. 11, a flow *management information* (B) is "systematized raw data" and a flow *management information* (C) is "processed data", a flow *management information* (D) is "management decision" and a flow *management information* (V) is "functional controls" in Fig. 11.

The functions (Steps) of the management transaction framework in Fig. 8 are classified and assigned to the EMC procedural elements: Interpretation (IN), Data Processing (DP), Decision Making (DM) and Realization of decisions (RE). In this way, four classes of Steps (meta-types of functions of management in Figs. 5 and 8) are specified: IN, DP, DM and RE. Thus, the meta-types of functions (Function 1, Function 2, …, Function n) of the managed activity framework (see Fig. 8) in the context of the EMC are depicted in

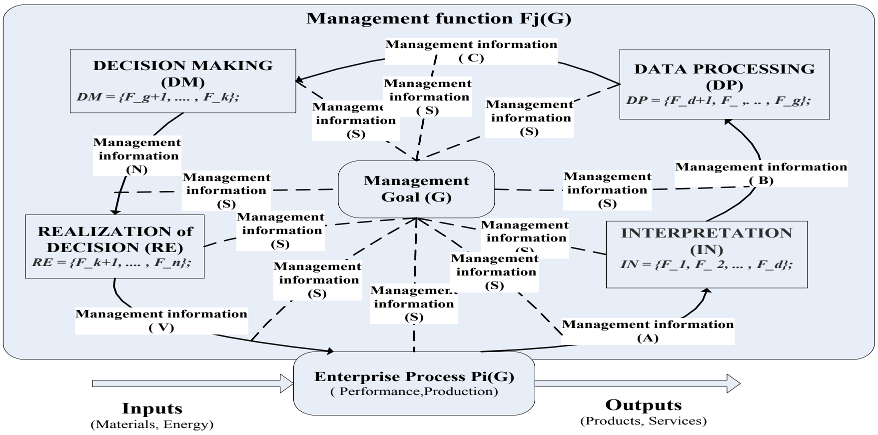


Fig. 11. The management transaction is defined as the elementary management cycle (EMC).

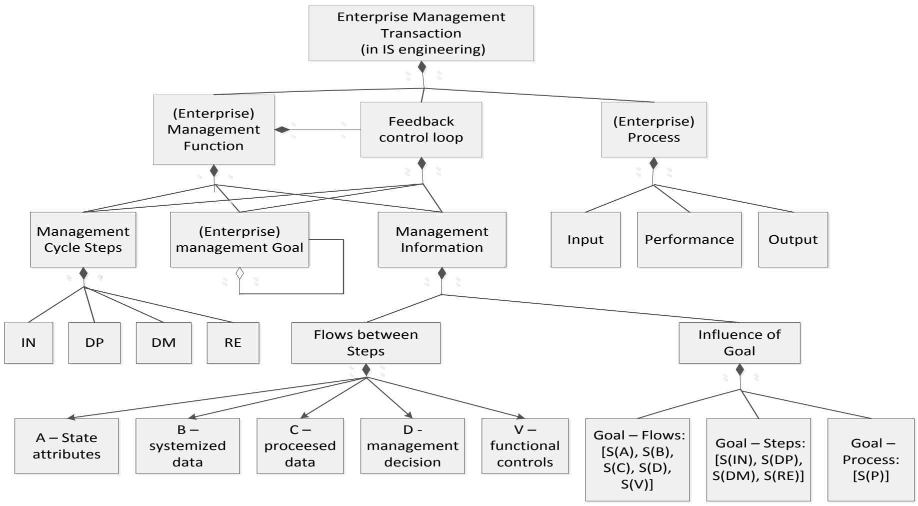


Fig. 12. A meta-model of the management transaction defined as the EMC.

Fig. 11, i.e. IN = $\{F_1, F_2, \ldots, F_d\}$; DP = $\{F_{d+1}, \ldots, F_g\}$; DM = $\{F_{g+1}, \ldots, F_k\}$; RE = $\{F_{k+1}, \ldots, F_n\}$.

The EMC steps (IN, DP, DM and RE) are the meta-types of *primary functions* common in business modelling, such as forecasting, planning and the rest in Fig. 7. Analysis of matching and mapping the *primary functions* and the EMC steps (IN, DP, DM and RE) is an interesting and relevant issue yet beyond the scope of this paper.

The set of management functions in Fig. 13 on the business management layer is associated with the Detailed Value Chain Model (Fig. 9) (Gudas and Lopata, 2002) and the components of any management function $F_j(G)$ on the IS engineering layer are defined by the elementary management cycle EMC (Figs. 11 and 12) (Gudas, 1991, 2012a; Gudas *et al.*, 2005).

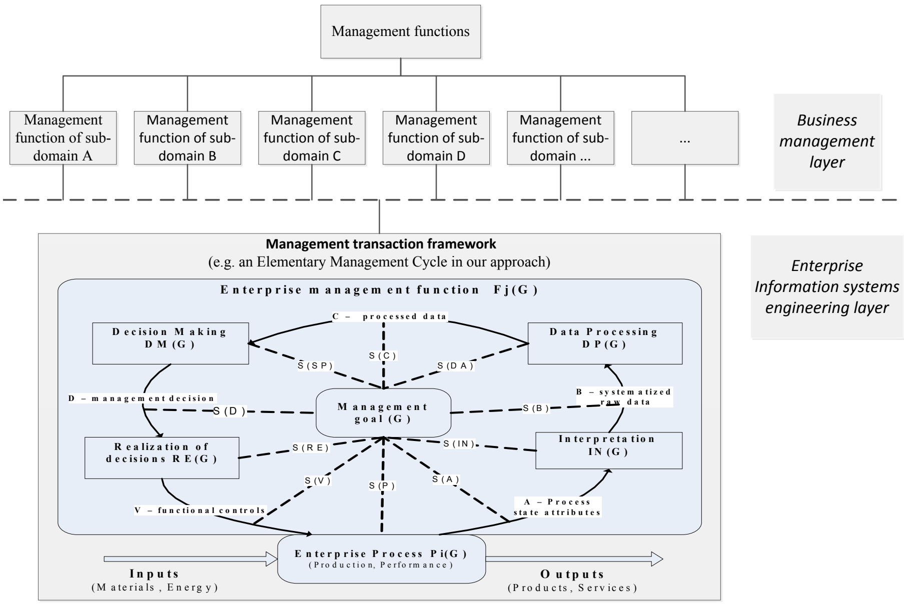


Fig. 13. A meta-structure of Enterprise management functions based on the EMC.

7. Conclusions

The lack of understanding of the enterprise IS application domain and domain knowledge is mentioned among the critical factors for enhancing IS development.

The emergence of new IS engineering methodologies and techniques is inspired by the application of the contemporary systems approach in the development of IS engineering techniques. Understanding of the IS application domain as a type of complex systems, self-managed systems is required for the advancement of MDA-based methods. The analysis of popular business modelling (BM) frameworks discovers the gap between the BM and enterprise (architecture) modelling (EM)/business process modelling (BPM) conceptual base. The problem is the consistence between EM and BPM approaches, i.e. these approaches lack certain modelling concepts that would reflect the essential characteristics of business enterprise management activities.

The presented analysis of the systemic characteristics of business modelling (BM) approaches (aiming at value creation) and modelling methods that aim at enterprise IS software development (EM, BPM, BPMM – BP management modelling) determines the difference of modelling paradigms to be implemented. The internal modelling paradigm (the white box approach) is naturally applied in business modelling (BM), whereas the external modelling paradigm (the black box approach) dominates in EM, BPM and BPMM methods. Business models for the original purpose – to manage organizational units and monitor internal activities and interactions – is by nature a result of internal modelling.

The internal view based approach to the analysis of the relationship between IS application domain models and business domain models is presented. The aim is to refine information transactions (hidden) in the business domain models meant for business management. The inherent characteristic of business models is complexity, i.e. business models are frameworks with a pre-defined semantic structure, for instance, the PDCA cycle, Value Chain Model, etc. Thus, the elementary building blocks of business modelling (BM) are structures, complex units (white boxes). The internal modelling viewpoint on EM and BPM methods and languages determines a diverse understanding of modelling concepts and, in particular, the topology of business models (BM) and the IS domain models in applications development methods. Meanwhile, the elementary building blocks of business process models (BPM) and Enterprise models (EM) are primitives, i.e. modelling concepts are specified as elementary units without any pre-defined structure. The external viewpoint common in ISE is unsufficient to model the essential management information interactions.

The common topology of models in EM and BPM as a matter of fact is “a set” of processes (or functions) or “workflow” of processes (or functions). That is, the EM and BPM modelling approaches are “process view” based (or “workflow thinking” based). Meanwhile, the essential topology of business models (BM) is characterized by the goal-driven feedback cycle of business activities (steps).

The methodology proposed to manage this inconsistency is based on the internal modelling paradigm: the IS application domain is defined as an organizational system – a *self-managed system* wherein the system’s goals influence the content of other elements of the management cycle. The lack of some important modelling constructs in EM and BPM is identified, e.g. the lack of concepts – *enterprise management function, management transaction, feedback control loop* – is the reason of disintegration of different layers of IS domain models and business models. The understanding of the concepts “function” and “management function” in business modelling and EM and BPM is different. The enterprise functions in business modelling (BM) approaches are considered from the external modelling perspective – detailed content of information is not a subject of BM. The management functions in EM and BPM approaches are complex components which include lower level elements (sub-functions and/or processes); however information transactions in EM and BPM are a subject of modelling from the external modelling perspective (Input, Process, Output). Some business process modelling approaches (BPMN, UML, IDEF3) include the concept *Types* for modelling *Events* (i.e. Event types are defined in the language notation), yet there are no *Types* defined for modelling *flow transformations*. For example, there are no *Types* defined for concepts *process, function, activity*, etc.

This gap of modelling concepts raises difficulties in the development of MDA-based techniques for consequential transformations of different modelling layers for the IS development needs. The metastructure of enterprise management transaction is defined and elaborated for business modelling and for IS application domain modelling. One of the alternatives for the internal IS domain modelling is the Detailed Value Chain Model (Gudas and Lopata, 2002, 2009) and the elementary management cycle (EMC) framework (Gudas, 1991, 2012a; Gudas *et al.*, 2005). Common BPM languages (BPMN, ARIS, etc.) are appropriate for representing DVCM and EMC, but should be complemented with constraints, based on the definition of the management transaction.

The presented analysis determines the limitations of "process-based" and "workflow-based" thinking as well as the limitations of process mapping techniques for business domain modelling for the IS development needs. An alternative way of the IS application domain modelling has been proposed, i.e. to switch to "management-control-transactions" thinking and management cycle based modelling techniques. The internal modelling based frameworks give a conceptual background for a knowledge-based BP and enterprise modelling for needs of IS engineering. The proposed frameworks should be considered as the predefined domain knowledge to be used in advanced modelling tools with a view to verify (validate) the application domain knowledge.

References

Ackoff, R.L. (1971). Towards a system of systems concepts. *Management Science*, 17(11), 661–671.

Andersson, B., Bergholtz, M., Edirisuriya, A., Ilayperuma, T., Johannesson, P. (2005). A declarative foundation of process models. In: *Proceedings of Advanced Information Systems Engineering: 18th International Conference (CAiSE05), Porto*, *Lecture Notes in Computer Science*, Vol. 3520. Springer-Verlag, Berlin, pp. 233–247.

Andersson, B., Bergholtz, M., Edirisuriya, A., Ilayperuma, I., Johannesson, P., Grégoire, B., Schmitt, M., Dubois, E., Abels, S., Hahn, A., Gordijn, J., Weigand, H., Wangler, B. (2006). Towards a reference ontology for business models. In: *Proceedings of the 25th International Conference on Conceptual Modelling, Tucson, USA, November 6–9*, pp. 482–496.

Avison, D.E., Fitzgerald, G. (2003). Where now for development methodologies. *Communications of the ACM*, 46(1), 79–82.

Barios, J., Montilva, J. (2004). Business modelling through roadmap. In: *Proceedings of the 6th International Conference on Enterprise Information Systems (ICEIS'2004), Porto, Portugal*, pp. 348–355.

Barker, R., Longman, C. (1992). *CASE Method: Function and Process Modelling*. Addison–Wesley, Boston.

Bergholtz, M., Jayaweera, P., Johannesson, P., Wohen, P. (2003). Reconciling physical, communicative, and social/institutional domains in agent oriented information systems – a unified framework. In: *Conceptual Modelling for Novel Application Domains*. *Lecture Notes in Computer Science*, Vol. 2814. Springer-Verlag, Berlin, pp. 180–194.

Bernus, P. (2001). Some thoughts on enterprise modelling. *Production Planning and Control*, 12(2), 110–118.

BMM (2008). *Business Motivation Model version 1.1 without change bars OMG*. Document number: formal/2010-05-01, standard document: http://www.omg.org/spec/BMM/1.1/.

BPDM (2008). *Business Process Definition MetaModel version 1.0 OMG*. Document number: formal/2008-11-03, standard document: http://www.omg.org/spec/BPDM/1.0.

BPMM (2008). *Business Process Maturity Model*. http://www.omg.org/spec/BPMM/.

BPMN (2011). *Business Process Model and Notation version 2.0 OMG*. Document number: formal/2011-01-03, standard document: http://www.omg.org/spec/BPMN/2.0.

Brown, P., Sondalini, M. (2004). *Developing an Asset Maintenance Quality Management System to ISO 9001 Requirements*. Lifetime Reliability Solutions, Perth, Western Australia. http://www.lifetime-reliability.com.au/free-articles/iso9001-quality-management-tips/maintenance-management-iso9001-quality-system.html.

Brun, Y., Serugendo, G.D.S., Gacek, C., Giese, H., Kienle, H., Litoiu, M., Müller, H., Pezzè, M., Shaw, M. (2009). Engineering self-adaptive systems through feedback loops. In: Cheng, B.H.C.,de Lemos,R., Giese, H., Inverardi, P., Magee, J. (Eds.), *Self-Adaptive Systems*, *LNCS*, Vol. 5525, pp. 48–70.

Chen, D., Doumeingts, G. (2003). European initiatives to develop interoperability of enterprise applications – basic concepts, framework and roadmap. *Annual Reviews in Control*, 27, 153–162.

Chen, D., Doumeingts, G., Vernadat, F. (2008). Architectures for enterprise integration and interoperatability: past, present, future. *Computers in Industry* 59, 647–659.

Cocchi, A. (2012). The emerging properties of business models: a systemic approach. In: Audretsch, D.B., Lehmann, E.E., Link, A.N., Starnecker, A. (Eds.), *Technology Transfer in the Global Economy. International Studies in Entrepreneurship*, Vol. 28. Springer, New York, pp. 277–299. doi:10.1007/978-1-4614-6102-9_15.

Davenport, T.H. (1993). *Process Innovation – Reengineering Work Through Information Technology*. Harvard Business School Press, Boston.
Deming, W.E. (1993). *The New Economics for Industry, Government and Education*. Massachusetts Institute of Technology, Center for Advanced Engineering Study, Cambridge.
DoD (1994). F*ramework for Managing Process Improvement. Appendix A*. Glossary Department of Defense.
DoD (2008). *Continuous Process Improvement/Lean Six Sigma Guidebook*. Revision 1, Department of Defense. http://www.au.af.mil/au/awc/awcgate/dod/cpi_leansixsigma_hdbk2008.pdf. Retrieved at 2014 02 19.
Dorf, R.C., Bishop, R.H. (2011). *Modern Control Systems*, 12th ed. Prentice Hall, Upper Saddle River.
Dubray, J.J. (2002). A novel approach for modelling business process definitions. *BPMN Working Draft*. http://www.ebpml.org/ebpml2.2.doc. Retrieved 2014 02 20.
ERM (2011). In: *Revised Papers Series: Lecture Notes in Business Information Processing*, Vol. 37, Springer, Berlin, 2009, pp. 91–102.
Frankel, D.S. (2003). *Model Driven Architecture: Applying MDA to Enterprise Computing*. Wiley, New York.
GERAM (1999). *Generalized Enterprise Reference Architecture and Methodology version 1.6.3*. IFAC-IFIP task force on architecture for enterprise integration. http://www.cit.gu.edu.au/~bernus/taskforce/geram/versions/geram1-6-2/v1.6.2.html. Retrieved at 2012.01.20.
Giaglis, M.A. (2001). Taxonomy of business process modelling and information systems modelling techniques. *The International Journal of Flexible Manufacturing Systems*, 13, 209–228.
Gordijn, J., Akkermans, H. (2003). Value based requirements engineering: exploring innovative e-commerce idea. *Requirements Engineering Journal*, 8(2), 114–134.
Gordijn, J., Akkermans, H., van Vliet, H. (2000). Business modelling is not process modelling. In: *Procedings of the 1th International Workshop on Conceptual Modelling Approaches for e-Business (eCOMO'2000), The 19th International Conference on Conceptual Modelling (ER'2000)*, Springer-Verlag, Heidelberg, pp. 40–51.
Gudas, S. (1991). A framework for research of information processing hierarchy in enterprise. *Mathematics and Computers in Simulation*, 33, 281–285.
Gudas, S. (2012a). *Foundations of the Information Systems' Engineering Theory*. Vilnius University, Vilnius (monograph, in Lithuanian).
Gudas, S. (2012b). Knowledge-based enterprise framework: a management control view. In: Hou, H.T. (Ed.), *New Research on Knowledge Management Models and Methods*, ISBN:978-953-51-0190-1, InTech. Available from: http://www.intechopen.com/books/new-research-on-knowledge-management-models-and-methods/knowledge-based-enterprise-framework-a-management-control-view.
Gudas, S., Lopata, A. (2002). Framework for identification of information resources. In: *Sbornik z Mezinarodni Conference "Nove Trendy Rozvoje Prumyslu", Brno, Česka Republika, 6–7 December, 2001*, Vysoke Učeni Technicke v Brne, Fakulta Podniklatelska, Brno.
Gudas, S., Lopata, A. (2009). Control view based elicititation of functional requirements. In: Abramowicz, W., Flejter, D. (Eds.) *BIS 2009 International Workshops "Business Information Systems Workshops", Poznan, Poland, 27–29 April, 2009*, *Lecture Notes in Business Information Processing*, Vol. 37. Springer, Berlin, pp. 91–102.
Gudas, S., Skersys, T., Lopata, A. (2005). Approach to enterprise modelling for information systems engineering. *Informatica*, 16(2), 175–192.
Harmon, P. (2010). The scope and evaluation of business process management. In: von Brocke, J., Rosemann, M. (Eds.), *International Handbooks on Information Systems, Handbook on Business Process Management*, Vol. 1. Springer-Verlag, Berlin, pp. 37–81.
Harmon, P. (2011). Capabilities and process. *Business Process Trends*, 9(13). http://www.bptrends.com/publicationfiles/07-12-2011-adv-capabilities%20and%20processes-harmon.pdf.
Henderson, J.C., Venkatraman, N. (1990). Strategic alignment: leveraging information technology for transforming organizations. *IBM Systems Journal*, 38(2–3), 472–484.
Hevner, A.R., March, S.T., Park, J., Ram, S. (2004). Design science in IS research. *MIS Quarterly*, 28(1), 75–105.
Humphrey, W.S. (1993). *Introduction to software process improvement*. Technical report CMU/SEI-92-TR-007, ESC-TR-92-007, Software Engineering Institute/Carnegie Mellon University.
ITIL (2007). *An Introductory Overview of ITIL V3 version 1.0*. Written by: Cartlidge, A., Hanna, A., Rudd, C., Macfarlane, I., Windebank, J., Rance, S. Published by: The UK Chapter of the itSMF. http://www.best-managementpractice.com/gempdf/itSMF_An_Introductory_Overview_of_ITIL_V3.pdf. Retrieved at 2014 02 17.
Johnson, R.A. (1976). *Management, Systems, and Society: An Introduction*. Goodyear Publishing Company, Pacific Palisades.

Kandjani, H., Bernus, P., Wen, L. (2012). Enterprise architecture cybernetics for complex global software development: reducing the complexity of global software development using extended axiomatic design theory. *Information Control Problems in Manufacturing*, 14 (1), 1233–1239.

Kandjani, H., Bernus, P., Nielsen, S. (2013). Enterprise architecture cybernetics and the edge of chaos: sustaining enterprises as complex systems in complex business environments. In: *System Sciences (HICSS), 46th Hawaii International Conference on System Sciences*, pp. 3858–3867.

Kephart, J.O., Chess, D.M. (2003). The vision of autonomic computing. *IEEE Computer*, 36(1), 41–50.

Korherr, B. (2007). *Business process modelling – languages, goals and variabilities*. PhD thesis, Institut für Softwaretechnik und Interaktive Systeme. http://publik.tuwien.ac.at/files/PubDat_141379.pdf.

Kramer, J., Magee, J. (2007). Self-management systems: an architectural challenge. In: *Future of Software Engineering (FOSE'07)*, IEEE, pp. 358–378.

Krogstie, J. (2012). *Model-Based Development and Evaluation of Information Systems: A Quality Approach*. Springer-Verlag, London.

Lankhorst, M., (2004). Enterprise architecture modelling – the issue of integration. *Advanced Engineering Informatics*, 18, 205–216.

Lankhorst, M., et al. (2005). *Enterprise Architecture at Work – Modelling, Communication, and Analysis*. Springer-Verlag, Berlin.

Leonard, M., Ralyte, J. (2010). From sustainable information system with a farandole of models to services. In: Nurcan, S., Salinesi, C., Souveyet, C., Ralyte, J. (Eds.), *Intentional Perspective on Information Systems Engineering*. Springer, Berlin, pp, 1–15.

List, B., Korherr, B. (2006). An evaluation of conceptual business process modelling language. In: *21st ACM Symposium on Applied Computing, Dijon, Frankreich, 04-23-2006–04-27-2006, Proceedings of the 21st ACM Symposium on Applied Computing, SIGAPP, Proceedings of the 2006 ACM Symposium on Applied Computing*, pp. 1532–1539.

Lopata, A., Gudas, S. (2009) Knowledge based refinement of business management functions. In: *Proceedings of 1st International Conference on Knowledge Engineering and Ontology Development (KEOD 2009), Funchal, Madeira, Portugal, 6–8 October, 2009*, pp. 435–442.

Lupeikiene, A., Dzemyda, G., Kiss, F., Caplinskas, A. (2014). Advanced planning and scheduling systems: modeling and implementation challenges. *Informatica*, 25(4), 581–616.

Medina-Mora, R., Winograd, T., Flores, R., Flores, F. (1992). The action workflow approach to workflow management technology. In: *CSCW 92 Proceedings*, pp. 281–288.

Moen, R., Clifford, N. (2014). *Evolution of the PDCA Cycle*. http://pkpinc.com/files/NA01MoenNormanFull paper.pdf. Retrieved 2014 01 27.

MOF (2014). *Microsoft Operations Framework 4.0*. http://www.microsoft.com/en-us/download/details.aspx?id =17647. Retrieved 2014 02 17.

Montilva, J.C., Barrios, J.A. (2004). BMM: a business modelling method for information systems development. *CLEI Electronic Journal*, 7(2).

Morkevicius, A., Gudas, S., Silingas, D. (2013). SBISAF: a service-oriented business and information systems alignment method. *Informatica*, 24(2), 231–251.

ODM (2009). *Ontology Definition Metamodel version 1.0 OMG*. Document number: formal/2009-05-01, standard document: http://www.omg.org/spec/ODM/1.0.

OMG Specifications (2014). *Business Modelling Specifications*. http://www.omg.org/spec/. Retrieved 2014 02 10.

OSM (2009). *Organization Structure Metamodel*. 2nd initial submission version 0.5.

Osterwalder, A., Pigneur, Y., Tucci, C.L. (2005). Clarifying business models: origins, present, and future of the concept. *Communications of the Association for Information Systems*, 15(5), 1–40.

Owens, J. (2013). *The Function Model: The Foundation for All Business Models*. http://www.orbussoftware.com/. Retrieved 2014 01 30.

Pavalkis, S., Nemuraitė, L., Butkienė, R. (2013). Derived properties: a user friendly approach to improving model traceability. *Information Technology and Control*, 42(1), 48–60.

Porter, M.E. (1985). *Competitive Advantage*. The Free Press, New York.

Rosenberg, D., Stephens, M., Collins-Cope, M. (2005). *Agile Development with ICONIX Process: People, Process, and Pragmatism*. Apress, New York.

Rummler, G.A., Ramias, A., Rummler, R.A. (2010). *White Space Revisited: Creating Value Through Process*. Wiley, San Francisco.

SARM (2010). *A Structured Approach to Enterprise Risk Management (ERM) and the Requirements of ISO 31000*. AIRMIC, Alarm, IRM. http://theirm.org/documents/sarm_final.PDf. Retrieved 2014 02 17.

SBVR (2013). *Semantics of Business Vocabulary and Business Rules version 1.1 OMG*. Document number: formal/2013-09-04, standard document: http://www.omg.org/spec/SBVR/1.1/PDF.

Scheer, A.W, Nuttgens, M. (2000). ARIS architecture and reference models for business process management. In: van der Aalst, W., et al. (Eds.), *Business Process Management*, *LNCS*, Vol. 1806. Springer-Verlag, Berlin, pp. 376–389.

Schekkerman, J. (2003). *How to Survive in the Jungle of Enterprise Architecture Frameworks*. Trafford, Bloomington. ISBN 141201607X.

Schwaber, K. (2004). *Agile Project Management with SCRUM*. Microsoft Press, Washington.

Skersys, T., Tutkutė, L., Butleris, R., Butkienė, R. (2012). Extending BPMN business process model with SBVR business vocabulary and rules. *Information Technology and Control*, 41(4), 356–367.

Skyttner, L. (2002). *General Systems Theory. Ideas and Applications*. World Scientific, Singapore.

Solvberg, A. (2010). On role of models in information systems. In: Nurcan, S., Salinesi, C., Souveyet, C., Ralyte, J. (Eds.), *Intentional Perspective on Information Systems Engineering*. Springer-Verlag, Berlin, pp. 17–38.

TOGAF (2009). *TOGAF version 9.1*. The Open Group Technical Standard.

Tripathi, P.N., Reddy, P.N. (2008). *Principles of Management*, 4th ed. McGraw-Hill, New Delhi.

Turban, E., Sharda, R.E., Delen, D. (2010). *Decision Support Systems and Business Intelligent Systems*. Prentice Hall, Upper Saddle River.

UML (2014). *Documents associated with Unified Modelling Language (UML), v2.4.1*. http://www.omg.org/spec/UML/2.4.1/. Retrieved 2014 02 11.

UPDM (2013). *Unified profile for the Department of Defense Architecture Framework (DoDAF) and the Ministry of Defense Architecture Framework (MODAF)*. http://www.omg.org/spec/UPDM/Current. Retrieved at 2014 02 17.

van der Aalst, W.M.P. (2013). Business process management: a comprehensive survey. *ISRN Software Engineering*, 2013. Article ID 507984, 37 pp. doi:10.1155/2013/507984.

van der Aalst, W.M.P., ter Hofstede, A.H.M., Weske, M. (2003). Business process management: a survey. In: *PM'03 Proceedings of the 2003 International Conference on Business Process Management*, Springer-Verlag, Berlin, pp. 1–12.

van Lamsweerde, A. (2001). Goal-oriented requirements engineering: a guided tour. In: *Proceedings 5th IEEE International Symposium on Requirements Engineering (ISRE)*, pp. 249–262.

Weske, M. (2007). *Business Process Management. Concepts, Languages, Architectures*. Springer-Verlag, New York.

Whitman, L., Ramachandran, K., Ketkar, V. (2001). Taxonomy of a living model of enterprise. In: *Proceedings of the 2001 Winter Simulation Conference*, pp. 848–855.

Wortmann, J.C., Hegge, H.M.H., Goossenaerts, J.B.M. (2001). Understanding enterprise modelling from product modelling. *Production Planning & Control*, 12(3), 234–244.

Wren, D.A., Bedeian, A.G., Breeze, J.D. (2002). The foundations of Henri Fayol's administrative theory. *Management Decision*, 40(9), 906–918.

Yolles, M. (2006). *Organizations as Complex Systems: An Introduction to Knowledge Cybernetics*. Information Age Publishing, Greenwich.

S. Gudas is a principal researcher and a head of Software Engineering Department at the Institute of Mathematics and Informatics (MII), and professor at the Department of Informatics of Kaunas Faculty of Humanities, Vilnius University, Lithuania. In 2005, passed the Doctor Habilitation procedure on the topic "Modelling of Knowledge-based Information Systems Engineering Processes". His main research interests include enterprise information systems knowledge-based engineering theory, knowledge-based enterprise modelling and software development. He is the author of monograph "Foundations of information systems engineering theory", author and co-author of more than 140 research publications.

A. Lopata is a professor and the head of Department of Informatics of Kaunas Faculty of Humanities, Vilnius University, Lihuania. His research interests include enterprise knowledge-based modelling, requirements engineering techniques and CASE tools. He is an author and co-author of more than 30 scientific publications, participated as the project manager (or member) in various scientific and business related projects.

Informacinių sistemų taikymo srities vidinis modeliavimas

Saulius GUDAS, Audrius LOPATA

Modeliais pagrįstos įmonių IS inžinerijos metodai apima kompiuterizuojamos srities (domeno) modeliavimo etapą. Šiame etape siekiama aprašyti esmines įmonių (t.y. organizacinių sistemų) charakteristikas. Verslo domeno modeliavimas yra santykinai atskira modeliavimo sritis, nukreipta vertei kurti, tačiau koreliuoja su IS taikymo srities modeliavimo metodais, gali suteikti naujų įžvalgų tobulinant organizacijų modeliavimo, veiklos procesų ir veiklos valdymo modeliavimo metodus. Veiklos srities (domeno) modeliavimo metodai IS inžinerijoje ir verslo domeno modeliavimo metodai nėra izoliuoti ir gali būti tiriami naudojant tą pačią modeliavimo paradigmą. Tačiau modeliais pagrįstame požiūryje, pastebimi tam tikri įmonės valdymo veiklų supratimo neapibrėžtumai. Problemiškas modeliavimo metodų nuoseklumas IS inžinerijoje nurodo domeno modeliavimo sąvokų sisteminės analizės poreikį. Taikant vidinio modeliavimo paradigmą, įmonės valdymo veikla analizuojama save valdančių sistemų požiūriu, taip paaiškėja, kad IS inžinerijoje iš dalies stinga konceptualaus pagrindo tokiam domeno modeliavimo požiūriui realizuoti. Pateiktas metodas yra skirtas atskleisti paslėptą organizacijų valdymo veiklos sąveikose informaciją. IS taikymo srities (domeno), kaip save valdančios sistemos supratimas, leido naujai apibrėžti tokias sąvokas kaip veiklos valdymo transakcija, valdymo funkcija ir veiklos procesas. Sukonstruotas veiklos valdymo transakcijos metastruktūros apibrėžimas verslo valdymo ir IS plėtros sluoksnių lygmenyse, pateikti pavyzdžiai.