Volume 10 • Issue 4 • 2018

sciendo

International Society for Manufacturing, Service and Management Engineering

received: 5 August 2018
accepted: 15 December 2018

pages: 34-54

Information-decision model for self-controlling enterprise processes

Mirosław Zaborowski

ABSTRACT
This study demonstrates that integrated management and direct control systems may be organised as integrated enterprise process control (EntPC) systems, which are composed of self-controlling enterprise business processes. A business process has been defined as a control system for business activities, which are considered to be business processes of the lower level, or as base processes that are control systems for control plants in the form of infrastructure operations. An enterprise process also influences its delivery. This definition has been generally compared with definitions used in approaches of BPMN, YAWL, ARIS, DEMO and MERODE. Each enterprise process has its own controlling unit that contains one information unit and one decision unit, as well as memory places of the information-decision state variables that are processed by the business transitions that belong to these units. The i-d state variables are attributes of business objects, i.e. business units, business roles, business activities, business accounts and business products. Their values are transferred between business transitions that belong to the same or different controlling units. Relationships between business objects, business transitions and i-d state variables, as well as the other most important concepts of the EntPC system framework (EntPCF), are presented in this paper as the class diagrams of the enterprise process control language (EntPCL).

KEY WORDS
enterprise integration, complex control systems, enterprise modelling, enterpriseprocess control, industry 4.0, enterprise architectures

10.2478/emj-2018-0021

Corresponding author:

Mirosław Zaborowski
WSB University,
Department of Management, Poland
e-mail: mzaborowski@wsb.edu.pl

Introduction

Integrated management and process control systems

From a technical perspective, any activity of an enterprise consists in performing business processes. Therefore, business process management that answers the questions "What should be done?" and "Where, when and how should it be done?" encompasses all areas of enterprise operations management. It includes production management, sales management, accounting, human resource management, and all other functions that correspond to modules of well-known enterprise resource planning (ERP) systems (Langenwalter, 2000). In manufacturing enterprises, according to the ISA-95 standard, an ERP system operates on the highest level of integrated management and process control systems (Sholten, 2007). The IMPC systems also include manufacturing execution systems (MES), supervisory control and data acquisition (SCADA) systems and programmable logic controllers (PLC) that belong to control levels of their functional hierarchy.

Control is generally defined as a goal-oriented action of an object, named a controlling unit, upon another object, named a control plant (Bubnicki, 2005). In the case of MES and SCADA systems, each of these objects is a system with a complex internal structure. Management, which essentially means the influence exerted over somebody to do something, is a special case of control. Thus, for every enterprise, its IMPC system may be regarded as a complex structure with one big controlling system and a control plant being a set of infrastructure processes, which are control plants of the base direct control systems at the PLC level. A controlling system may be perceived as a central controlling unit of the entire IMPC system.

Information-decision state of IMPC controlling systems

IMPC systems are multilevel discrete-time control systems. This means that information processing is allowed only at discrete-time instants that are separated by discrete-time periods, whose length depends on the organisational level. Discrete-time periods and their end instants are identified by pairs

$$(l,t) \in Tl \subset L \times T,$$

in which the identification numbers of time instants, $t \in T$, obtain integer values from the sets assigned to their time scale numbers, $l \in L$. In management subsystems, discrete-time periods are often referred to as planning periods.

Every IMPC system is an IT system. Thus, information flow in an IMPC system consists in recording data to the memory of its controlling system and reading it at the same discrete-time instant or at a later moment in time. The information-decision state of an IMPC system at a given discrete-time period

$$x_{l,t} = [v_{l,t}, u_{l,t}], \qquad for\ (l,t) \in Tl, \qquad (1)$$

is a set of values of the i-d state variables, $x_{ih}(l,t)$, that are assigned not only to the instants (l,t), but also to the instants $(l, t+h)$, shifted in time, back or forward, by a definite number h of discrete-time periods. The i-d state represents all current and past information as well as forecasts and decisions concerning the future, that are recorded in the memory of the controlling system and are needed to make new decisions. They are introduced to the controlling system from outside by its users and by measurement devices as external input variables

$$u_{l,t} = f^{ext}(l,t), \qquad for\ (l,t) \in Tl, \qquad (2)$$

or they are calculated in the controlling system at the beginning of the discrete-time period, as internal i-d state variables

$$v_{l,t} = f((l,t), x_{l,t-1}, u_{l,t}), \qquad for\ (l,t) \in Tl \quad (3)$$

Inputs to the procedures performed in the controlling system are external input variables $u_{l,t}$ and preceding i-d state variables $x_{l,t-1}$.

One should note that external input variables are the output variables of procedures that introduce data to the controlling system. Thus, all i-d state variables are output variables of procedures performed in the controlling system.

The model (1)(2)(3) is correct under the assumption that introducing and processing data durations at the beginning of discrete-time periods may be neglected in comparison with the duration of these periods. Therefore, one should show how to organise those data processing procedures that do not satisfy this assumption (Zaborowski, 2018).

From an IT point of view, equation (3) is a static model of the cause-result dependencies between the input and output variables of procedures that are performed at settled discrete-time instants (l, t). However, from the control theory perspective, it is also a dynamic model of the IMPC controlling systems because the coordinates of the i-d state vector $x_{l,t}$, which are not visible in vector equations (1)(2) and (3), are i-d state variables $x_{ih}(l, t)$ that are assigned not only to the instants (l,t) but also to the instants shifted in time $(l, t+h)$.

Enterprise reference architectures

The general mathematical model of IMPC controlling systems in the form of difference equations may facilitate transferring the results of the classical control theory to the systems of enterprise management, e.g., to analyse enterprise stability and controllability or to assess management quality using criteria and methods applied to the control systems. However, practical conclusions from such an analysis always concern i-d state variables perceived as attributes of enterprise processes or attributes of structural objects that belong to these processes. Therefore, any theory describing the structure and operation of all IMPC systems must include a universal model of the structure of the enterprise processes as well as the structure and functioning of their control systems and interactions between them. The author's theory of Enterprise Process Control (EntPC theory) (Zaborowski, 2016a) satisfies these requirements.

Specific IMPC systems, whose structure conforms with the EntPC reference model, have been named Integrated Enterprise Process Control (EntPC) systems. In an EntPC system, control is decentralised in the hierarchical organisational structure, where control plants may be subordinate control systems (Mesarović, Macko & Takahara, 1970), and in the multistage structure of transactions (Dietz, 2006a,b) between delivery and receiving processes (Fig. 1). Such a decentralised system includes production, preparatory and managerial processes that interact on all organisational levels of an enterprise.

levels. Therefore, in the case of the EntPC theory, the common model of enterprise processes on the ERP, MES and SCADA levels was inspired by standards for modelling business processes (van der Aalst & van Hee, 2002; Reijers, 2003; Dietz, 2006a; Davis & Brabander, 2007; Hofstede et al., 2010; Weske, 2012; BPMN, 2013; WFMC, 1999). Furthermore, it was assumed that the feedback structure should be adopted not only to individual process control systems on the PLC level but also to all individual control systems of enterprise business processes.

I-d state variables are not only attributes of enterprise processes but also attributes of other structural objects that belong to IMPC systems. Specifications of those structural enterprise objects that are relevant to the construction and functioning of IMPC systems, as well as their relationships, are accessible as parts of different "enterprise reference architectures", also called "enterprise architecture frameworks" (EAF), (Bernus, Noran & Molina, 2015; Kosanke, Vernadat & Zelm, 1999; Noran, 2003; Panetto, 2007; Saha, 2004; Vernadat, 2002; Williams, 1994). Similar specifications are the content of class

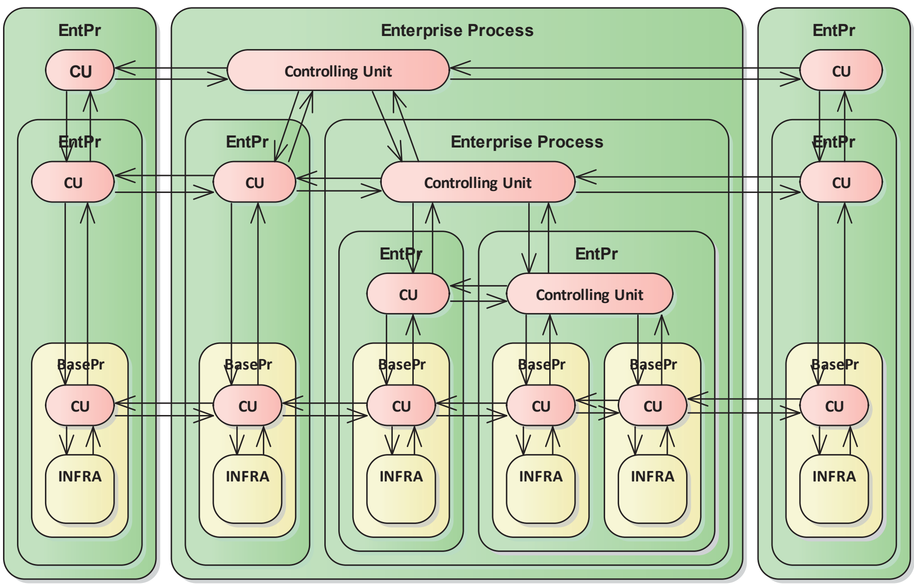

Fig. 1. Sketch of hierarchical and transactional couplings between control systems of business and base processes

Process control on the MES and SCADA levels, just as business process management on the ERP level, answers the questions "What should be done?" (what products and how many/ much of them) as well as "When and how should it be done?" (at which discrete-time periods and with what values of quality parameters). The question "where" (in which organisational units) is relevant on the ERP and MES

diagrams for enterprise conceptual models, which are metamodels of graphical languages for modelling enterprise architectures, e.g. ArchiMate (Iacob et al., 2012) and UEML (Vernadat, 2002; Panetto, 2007). The EAF's are the fundamentals of standards for IMPC systems. For example, the ISA-95 standard is based on the Purdue Enterprise Reference Architecture (PERA) (Williams, 1994).

In the case of EntPC systems, the metamodel of the Enterprise Process Control Language (EntPCL) is a part of the Enterprise Process Control Framework (EntPCF). The EntPCF defines how to create and use specific EntPC systems. The thesis on the generality of the EntPC theory (Zaborowski, 2016a) says that every IMPC system, irrespective of the enterprise, in which it is implemented, may be replaced, retaining all its functions and data, with a corresponding EntPC system, whose structure is conformable with the EntPCL metamodel. The EntPC theory includes the EntPCF itself and the thesis on its generality, as well as arguments and case studies, demonstrating that the thesis is true. The abbreviation "EntPC" is used instead of "EPC" to avoid confusion with the acronym of Event-driven Process Chain (Davis & Brabander, 2007).

The concepts of the EntPC theory are defined deductively, beginning with the most general concepts and moving forward, step by step, to those related to the structural details of the EntPC systems. The first part of the EntPC theory includes the information-decision model of enterprise business processes, which is presented in this paper.

1. Business processes

1.1. Definitions

Based on the APICS dictionary (Blackstone & Cox, 2005), a "business process" is a set of logically related tasks or activities performed to achieve a defined business outcome. According to van der Aalst & van Hee (2002), "a business process is one focused upon the production of particular products. These may be physical products. The 'product' can also be a service.". Thus, business processes are divided into manufacturing processes, service processes and administrative processes. Administrative processes, whose products are documents, may be listed among service processes (Reijers, 2003).

In the previous versions of the EntPC theory, that is in the ERC and EPC2 theory (Zaborowski, 2009, 2011), the business process was defined as an ordered set of activities and related resources. Resources are used, consumed or produced by these activities. In the EntPC theory, product places, which are found among "business accounts", as well as business products themselves, are components of business activities.

Resources, both consumable and reusable, are the means needed to perform business processes (Reijers, 2003). In the EntPC theory (and in its previous versions) resources are passive objects, although in the YAWL standard (Hofstede et al., 2010) and in all cases when business processes are administrative, they are regarded as actors performing business tasks or activities (Tab. 1).

For every modelling standard, a business process includes not only business activities but also certain elements that are designed to control their execution (Badura, 2014). They are events and gateways for BPMN; events and rules for ARIS; events, conditions and task decorators for YAWL; and selected business transitions (e.g., Boolean transitions) and guard conditions for EntPCL (Tab. 1).

The terms "business process" and "business activity" may be understood, depending on the context and on the accepted convention, as a model or as an instance of a given process or activity (Weske, 2012). According to the EntPC theory, these terms are assumed to refer to the models, whereas the specific realisations of business processes and activities are referred to, respectively, as business works or cases and business tasks (Tab. 1). A business task is a single or serial execution of a business activity. The duration of business tasks is usually equal to or greater than one discrete-time period, whereas durations of business events, which are business transition executions, formally are equal to zero.

In the EntPC theory, business processes are subdivided into production, preparatory and managerial processes. They are analogous to the primary, secondary and tertiary processes that have been described by van der Aalst & van Hee (2002). Production-oriented processes are defined as production processes (including manufacturing and service processes) and preparatory processes (repairs, overhauls, tooling setups and the like).

Different approaches to modelling enterprise business processes (Tab. 1) provide different meanings of the term "business process". In the DEMO (Design and Methodology for Organisations) (Dietz, 2006a,b) and EntPCL approaches, business processes are production-oriented. In the MERODE method of enterprise modelling for enterprise information system engineering (Snoeck, 2014), business processes are managerial workflow processes, which influence business objects through information system services and business events. In this approach, all production-oriented processes are counted among business objects, not among business processes. In the BPMN (BPMN, 2013), YAWL (Hofstede et al., 2010) and ArchiMate (Iacob et al., 2012) approaches, business processes are described as administrative processes. Manufacturing activities may belong to them as "waiting" activities or "manual" activities that are performed outside the modelling system.

Tab. 1. Selected concepts of the EntPCL and similar concepts used in the well-known approaches to business process modelling

EntPCL	DEMO	ArchiMate	YAWL	BPMN	ARIS	MERODE
business process	business process					
managerial business process			YAWL net	process	process	business process
business activity	P-act type					
managerial activity		business process	task	activity	function	service task
business object	fact type	business object	data element	data object (model)	entity	business object type
business product	P-fact type	product / business service	document type	document (model)	product/ service	
business transition	transition C-act type	business event application component	event	event (model)		event type
guard condition			condition		event	
variable		data object	variable	property	attribute	attribute
Boolean transition			task decorator	gateway	rule	
business unit	actor	business actor	resource	participant	organisational unit	
business role	actor role	business role/ business function	resource role	partner role	position	
business work	process performance					
managerial work			case	process (instance)	process occurrence	
business task	P-act					
managerial task			work item	activity (instance)	function occurrence	
realisation object	fact		case data element	data object (instance)	entity occurrence	business object
business task product	P-fact		case document	document	product/ service occurrence	
business event	event		event log	event	event occurrence	event
variable record			data value	property (instance)	entity occur. attribute	

1.2. Production-oriented business processes

The first axiom of the PSI theory (Performance in Social Interaction), which underlies the DEMO methodology (Design and Engineering Methodology for Organisations), states that "the operation of an enterprise is constituted by the activities of actor roles: … production acts (P-acts) and coordination acts (C-acts). These acts have definite results: production facts (P-facts) and coordination facts (C-facts)" (Dietz, 2006b). Therefore, the active and passive objects listed in Tab. 1 may be regarded correspondingly as acts and facts or as their types.

Based on the PSI theory, a business process called herein a multi-transactional business process, is "a process that consists of an ordered collection of transaction types" (Dietz, 2006b). A transaction is a sequence of acts that includes one P-act and several C-acts, which are the activities of the two actor roles: initiator and executor of the transaction. Thus, all C-acts of a transaction are assigned to the P-act performed by its executor.

According to the EntPC theory, a business process is a self-controlling business process defined as a system of control for a finite, partially ordered set of business activities that transform material resources, documents and services into products to fulfil the requirements of other business processes, that belong to a given enterprise or to its environment. Output products, just as input products, may be material resources, services or documents. A service is an outcome of a service process. Business activities, $a \in A$, are stages of business processes, $p \in Pa$. Conversely, every business process observed from the outside is a business activity. For example, the process F is an activity in the process X (Fig. 2). Therefore, the set of business processes is a subset of the set of business activities, $Pa \in A$. A business activity is a business process of a lower level or a base process that has no subordinate activities. A base process,

$$p \in Pb = Ab \subset A,$$

is a system that controls an infrastructure process.

Strictly speaking, the equal sign in the above formula does not concern the sets of base processes and activities. It relates to the sets of their identification numbers, but in practice (and in all formulas of the EntPC theory), software objects (representing real objects) are equated with their identifiers.

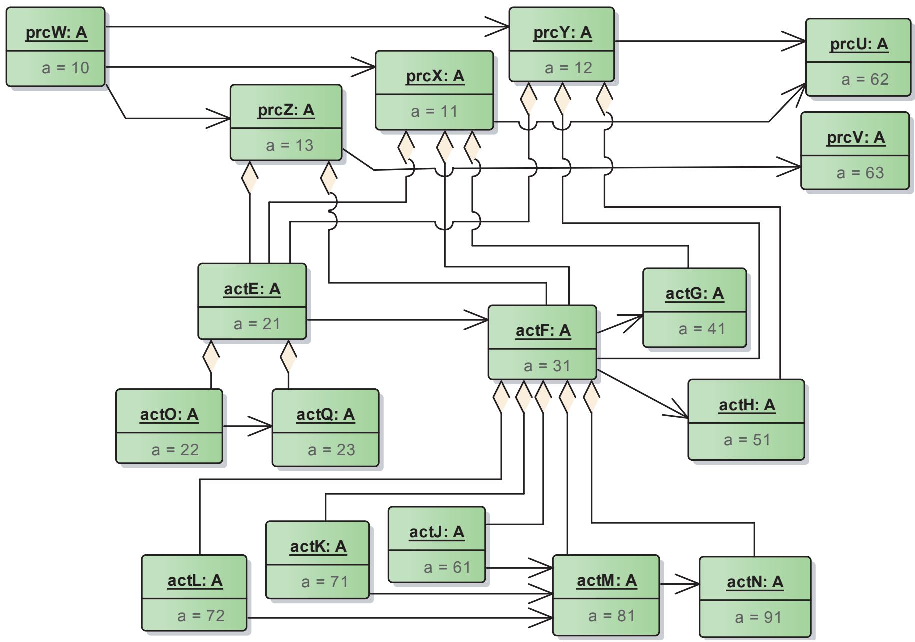

Fig. 2. Example of an EntPCL object diagram for activities in business processes

Unlike activities and processes, which are active objects, all business products, including services, are passive objects. The output products of business activities, after withdrawal to other business activities, formally become their input products, which are other structural objects (Fig. 3).

Each self-controlling business or base process has exactly one controlling unit that performs all its C-acts. If it is a base process, then it has only one stage consisting of only one P-act type that is an infrastructural process. If it is a business process, then

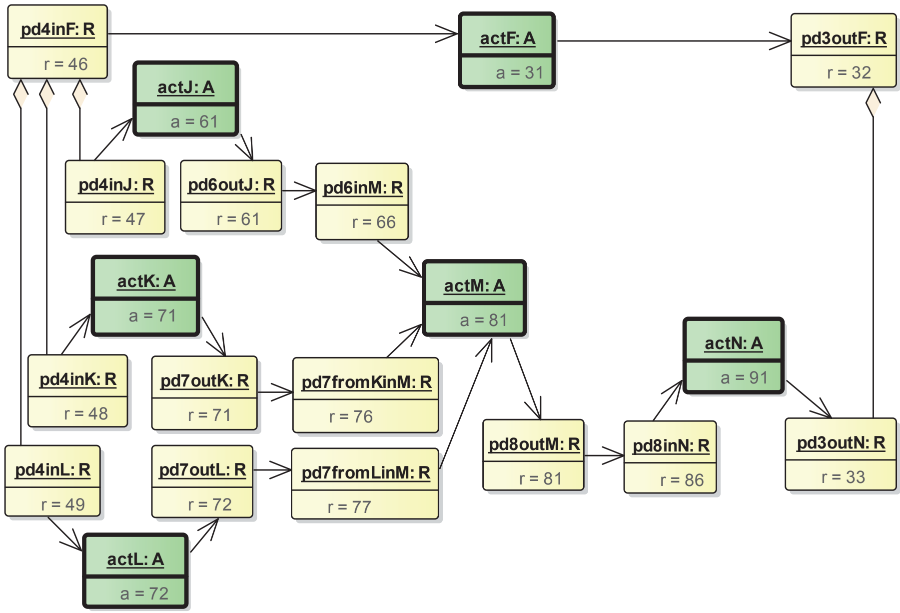

Fig. 3. Example of an EntPCL object diagram for the flow of products in business processes

its stages are business activities, which are P-act types whose components are C-act types belonging to the controlling units of the corresponding subordinate processes.

It is easy to notice that all business processes, irrespective of their definitions, are ordered sets of act types, whose common purpose is making products. However, in the case of administrative processes, which are often equated with business processes (because of the popularity of the BPMN standard), P-acts have no special role in the process structure. In contrast, in the case of self-controlling business processes, as well as for multi-transactional business processes, all C-acts are assigned to definite P-acts.

The structure of self-controlling business processes determines the following:

- hierarchical relationships between business activities and business processes,
- order relationships of performing business process activities,
- relationships between business activities and their input and output products,
- relationships between output activity products and input products of the following activities.

All these relationships may be shown in tabular form or in EntPCL diagrams (Zaborowski 2015, 2016a). They are patterned on UML object diagrams (Booch, Rumbaugh & Jacobson, 1999), but they are simpler because the only relationships between the presented objects are associations. Among associations, just as for the UML standard, one can separate the composition and weak aggregation. Additionally, the order relationships, represented by arrows, are distinguished. These relationships present the sequence of a performed activity (Fig. 2), the flow of products between activities (Fig. 3), the sequence of events, the information flow between software objects and the like. Therefore, just as for ArchiMate (Iacob et al., 2012), other diagrams illustrating a sequence of events or activities, e.g. UML activity diagrams, are not needed.

In the EntPCL diagrams, only those object attributes are visible that are useful for describing relationships between objects. Object identifiers are presented as the first of these attributes. Furthermore, they are the identification numbers of rows in the tables of enterprise objects belonging to their classes.

Sometimes, the same activity is a stage in different business processes. In other words, the same business activity may be an element of different business processes. For example, activity E belongs not only to process X but also to processes Y and Z (Fig. 2). Such relationships are presented as elements of a weak aggregation relation.

Every self-controlling business process has its own controlling unit that controls its subordinate and delivery activities. In the illustrative object diagram (Fig. 2), the unit that controls activities E, F and G and belongs to the process X is hidden in the activity denoted by ‘prcX’. Similarly, the controlling unit that controls activities J, K, L, M, and N of the process F is hidden in the activity 'actF'. The controlling unit of a specific activity determines its decisions based not only on superordinate decisions but also on requirements that are submitted by the controlling units of the receiving activities (e.g., the controlling unit of the activity F reacts to decisions from the controlling units of processes X, Y and Z as well as to orders from the controlling units of activities G and H). One should remember that the concepts "activities" and "processes" do not denote their concrete realisations (Tab. 1). They are understood as models that concern all feasible instances of their structural relationships.

1.3. Business units

Each specific business activity, $a \in A$, is performed by only one business unit, $u \in U$. This fact is formally modelled by a composition relationship, in which the business activity is a component of a specific business unit (Fig. 4). Similarly, each business process, $p \in Pa$, is a component of a definite business system, $s \in S$. These relationships are function dependencies:

$$u = u(a) \in U, for \;\; a \in A$$

$$s = s(p) \in S, for \;\; p \in Pa$$

On the other hand, there is exactly one generic activity, $a^n(a) \in An$, that is attributed to each business activity.

Business systems and business units are defined as systems of control, correspondingly for all the business processes and business activities that are performed by these systems and units. If a business system is not an elementary one, its controlling unit also controls business units that belong to it. Conversely, business systems, if observed from the outside, are business units of a higher level, $S \subseteq U$. In the structure tree of business units, the enterprise as a whole is the root. Business units, except the ones from the highest level, are components of business systems (Fig. 4) $s = s(u) \in S$.

The hierarchical structures of business systems correspond to analogous structures of business processes (Figs. 2, 3). The difference between these structures is that relationships between business systems and units are composition relationships, whereas, between business processes and activities, they are weak aggregation relationships.

Among all business systems of an enterprise, organisational systems, $s \in Sorg \subset S$, are particularly notable. Typical examples of organisational systems that belong to different organisational levels include the following:

- the enterprise as a whole,
- a work site,
- a workshop, and
- a workstation.

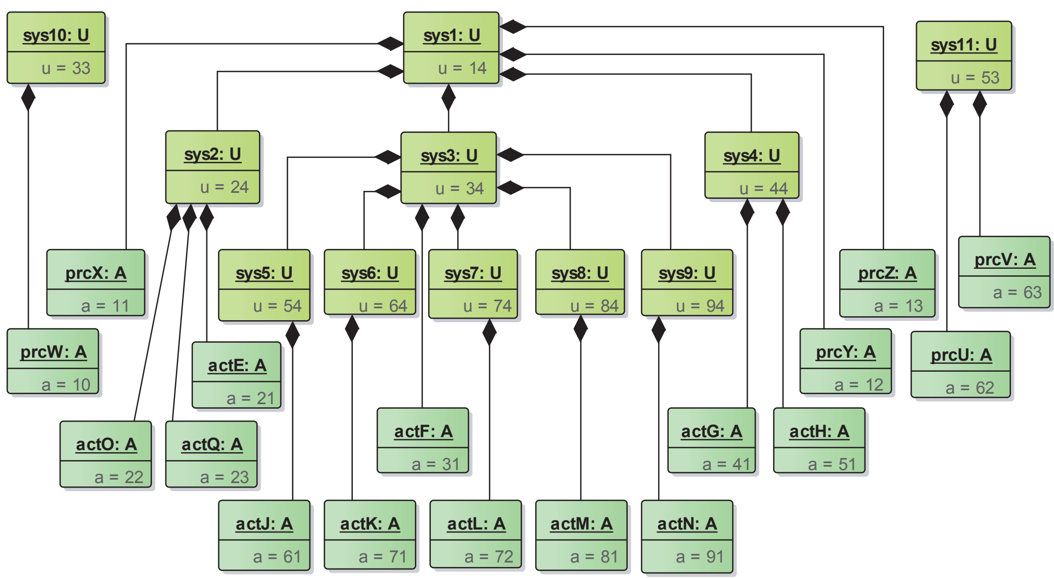

Fig. 4. Structure tree of the business units and business activities of an illustrative EntPC system

Organisational systems, if watched from the outside, are work units, $Sorg \subseteq Uw \subset U$, which are components of organisational systems of a higher level.

Among business units, we distinguish not only work units, but also operational segments and executive sectors. An operational segment, $u \subseteq Uo \subset U$, is a set of parallel work units that belong to the same organisational system and are grouped by similar products that can be mutually substituted. Work units may also be grouped if they share input resources, if their production technologies are similar and the like. Examples of operational segments include the following:

- departments of a work site (i.e., groups of its workshops) and
- work centres of a workshop (i.e., groups of its workstations).

An executive sector, $s \in Sh \subseteq Uh \subset U$, is a subset of operational segments that belong to the same organisational system, which is designed to execute managerial, preparatory or production subprocesses of work processes performed by this system. An organisational system regarded as a system that controls executive sectors is named a work system, $s \in Sw \subseteq Uw \subset U$. Controlling units of

- work systems,
- executive sectors and
- operational segments,

which belong to organisational systems of the same organisational level, form three functional layers of the executive, coordinative and allocative management in this level.

An enterprise is an independent organisational system that is involved in the production of goods or services to satisfy the requirements of consumers or other enterprises. According to the EntPC theory, every organisational system of an enterprise environment is regarded as a supplier or receiver of specific goods or services. The enterprise with its environment corresponds to the complete EntPC system. It is an integrated system that manages all business processes and controls all infrastructural processes in a given enterprise and in its business environment. Including the enterprise environment in the EntPC system causes influences from its outside processes to not be considered.

1.4. Enterprise processes

In EntPC systems, business units, $u \in U$, and business systems, $s \in S$, are identified correspondingly by the identifiers of the groups of all business activities, $a \in A$, and all business processes, $p \in Pa$, performed by them. Therefore, as collected business activities and collected business processes, they are counted among generalised business activities, $b \in B$, and generalised business processes, $p \in P$.

The business role, $g \in G$, of the business unit, $u(g) \in U$, is a group of jointly managed business activities distinguished in this unit according to the required competence and authority or required resources. The business system role, $p \in Pg \subset P$, is an analogously defined group of business processes. As a control system, the business system role controls its business processes and the roles of subordinate business units. By analogy to business units and business systems, their roles are also counted among generalised business activities and processes.

Enterprise processes are generalised business processes including business processes, $p \in Pa$, business systems, $s \in S$, their roles, $p \in Pg$, and also base processes, $p \in Pb$,

$$p \in P = Pa \cup S \cup Pg \cup Pb$$

Analogously, enterprise activities are generalised business activities,

$$b \in B = A \cup U \cup G$$

2. Control in enterprise processes

2.1. Business processes and base processes as control systems

The classical structure of a simple system of direct control (Murril, 2000) is a feedback loop consisting of a control plant, a measurement device, a controller and an actuating device. In the simplified form, measurement and actuating devices are hidden in a control plant (Bubnicki, 2005). However, the structure of a feedback control system may also be presented in another way (Fig. 5), facilitating its use for describing business process control systems. In this structure, only the actuating device is included in a control plant, whereas a measurement device is presented as a part of the controlling unit. In such systems, control encompasses the following:

- acquisition of information on a control plant and
- making decisions concerning the control plant.

Thus, the controlling unit is a composition of the information unit and the decision unit.

In a general case, controlled processes are multivariable control plants. Thus, in a self-controlling base process, which is a system of direct control with an infrastructural process as a control plant, the base information unit and the base decision unit are the corresponding collections of measurement devices and controllers. A self-controlling business process has the same structure (Fig. 5), but the internal elements of the information and decision units are different. Moreover, its control plant is a set of business activities that may be subordinate, base or business processes. Consequently, the EntPC system of an entire enterprise may be presented as a functional block diagram that includes only controlling units and infrastructural control plants (Fig. 1).

The output variables of information and decision units are referred to as information variables and decision variables, respectively. The base controlling variables affect infrastructure control plants through their actuating devices. Decision variables, in this case, are supervisory setting variables. Moreover, information variables are measured controlled variables and measured disturbances. In the case of a simple automatic control system, the base decision unit is a controller, and the base information unit is a measurement device.

The exact values of controlled variables in a base control system are not known. They differ from corresponding information variables because of measurement errors. However, everybody knows that controlled variables depend on base controlling variables and on disturbances that arise from the infrastructural environment of the control plant. Consequently, measurements of controlled variables depend on both these variables and measurement errors. The measured values of controlled variables and selected disturbances are accessible as output variables of a base information unit.

2.2. Business transitions and business events

Information units and decision units are functional units of enterprise processes, $k \in Kf$. Every functional unit is an information-decision action, $k \in Kf \subset Kid$, that is a business transition, or an information-decision process watched from the outside (Fig. 9). Business transitions, $k \in K \subset Kid$, are elementary software objects that are designed for processing information and decisions. An information-decision process, $k \in Pid \subset Kid$, is an

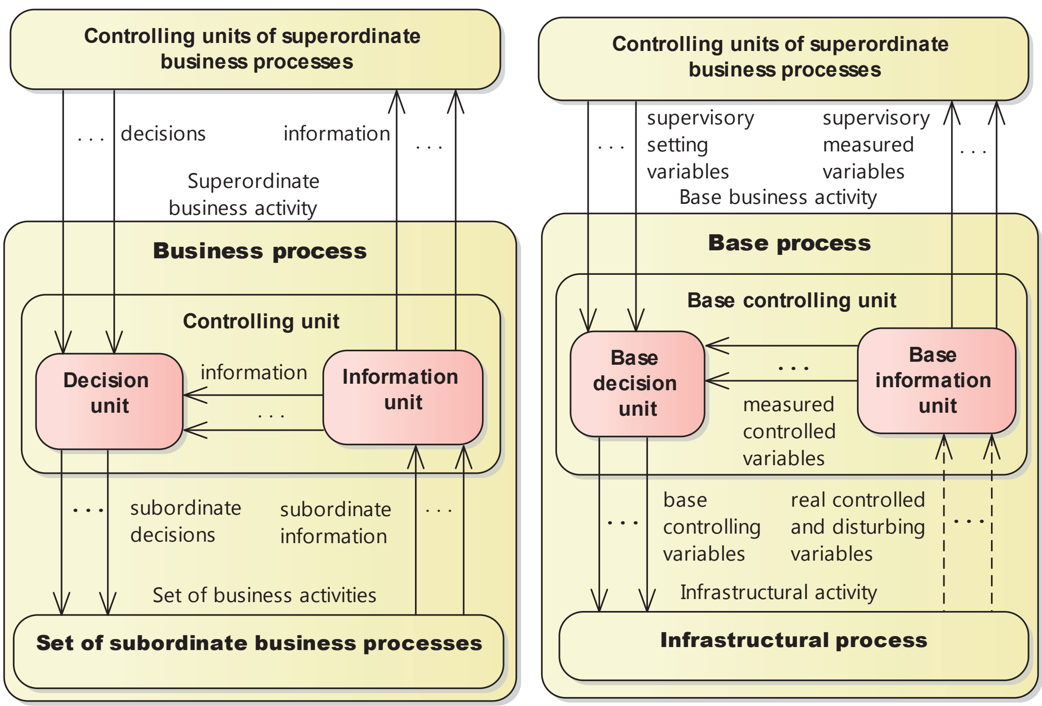

Fig. 5. Business and base processes as control systems

ordered set of business transitions, which can be executed at the same discrete-time instant. Each business transition belongs to a definite functional unit, a definite controlling unit, and consequently, a definite enterprise process and a definite enterprise activity,

$$p(k) \in P \subseteq B, for \ k \in K$$

Every business transition, regarded as a software object, has exactly one business operation (Fig. 9), which is an operation that processes its input variables into output variables. It has also one operation for reading input variables and one operation for recording output variables. Additionally, when an EntPC system moves to the next discrete-time instant, then business transitions shift their i-d state variables, that are their output variables. In EntPC systems, business transitions are the only software objects that perform data processing operations. All other objects of these systems, except for clocks initiating sequences of business events, have only writing and reading operations.

The i-d state variables are input and output variables of business transitions. They are passive objects of business transitions environment through which they can communicate. Guard variables and guard conditions are the other inputs and outputs of business transitions. They are binary variables used to control executions of the transitions. Guard variables, like i-d state variables, are attributes of structural objects, whereas guard conditions are attributed to the business transitions. Some business transitions, corresponding to events and gateways of the BPMN standard (Tab. 1), process only guard conditions and guard variables, but in a general case, they can perform the more complex procedures of data processing, e.g. the algorithm of digital PID controllers in direct control systems (Murril, 2000), the MRP algorithm (Orlicky, 1975) for ERP systems and the like.

Business transitions that belong to information and decision units are referred to as information transitions, $k \in Ki \subset K$, and decision transitions, $k \in Kd \subset K$, respectively. Their output variables are the same information and decision variables, which are the output variables of functional units (Fig. 6).

Business transitions are coupled with their input and output variables, which are passive objects, and do not have any direct couplings. Transitions stimulated by clock impulses at consecutive discrete-time instants investigate the states of variables in their environment in order to decide whether to begin their operations. In this sense, business transitions are autonomous software objects (Lockemann, 2006). Consequently, EntPC systems may be counted among multiagent control systems with passive interactions between agents (Monostori et al., 2015).

According to the EntPC theory, every business event, $e \in E$, is an execution of a definite business transition operation. The duration of each business operation is formally equal to 0, and all the operation is attributed to a concrete discrete-time instant. An EntPC system works properly, if the following requirements, which are axioms of the EntPC theory, are satisfied:

- first, the duration of performing a business operation is so short that the interval between the initial moment of the discrete-time period and the end moment of the operation is imperceptible relative to the length of this period;
- second, none of the business transitions can act in a given discrete-time period more than once, to enable attributing one definite value to a given i-d state variable at a given discrete-time period;
- third, in a given discrete-time instant, the business transition must act according to a definite order, to avoid casual variations of the i-d state variables.

Therefore, one part of the EntPC theory is devoted to explaining how to organise the managerial activities that replace those business transitions that do not satisfy the first of the above axioms and how to assure fulfilment of the second and third axioms (Zaborowski, 2018).

2.3. Hierarchical and transactional couplings between enterprise processes

In EntPC systems, information flows consist in recording values of variables determined by business transitions to memory places in the controlling units of individual business processes and then of reading by other business transitions. Information variables are remembered in the same controlling unit that includes recording information transitions (Fig. 6). They are read

- as controlled variables by a decision unit,
- as subordinate information variables by information units of superordinate business processes,
- and by other transitions of the same information unit. Furthermore, information transitions can generate

- disturbance information variables, as illustrated by the dashed lines in Fig. 6 that carry information on disturbances measured in a given process.

Unlike information variables, the decision variables of EntPC systems are kept not in the controlling units, from which they come but in the controlling units that include the decision transitions that read their values. To clarify, decisions are remembered where they are to be executed. Decision variables are recorded as follows:

- as superordinate decision variables by the decision units of superordinate business processes,
- as subordinate decision variables that are recorded by the decision units of business processes in the decision units of subordinate processes; they may also be recorded by the decision units of base processes in the memory places of the base controlling variables (Fig. 5),
- as order variables by the decision units of receiving processes,
- as cooperative variables by the decision units of delivery processes that transfer information on the products available for reception (analogously, transfer variables from a given process are recorded in the controlling units of receiving processes),
- as return transfer variables by the decision units of receiving processes that transfer information on received business products (in particular, on the products that have been rejected),
- as return order variables by the decision units of delivery processes that transfer information on offers and rejected orders for delivery products, and
- as external disturbance variables that carry information on disturbances measured and made available in other processes.

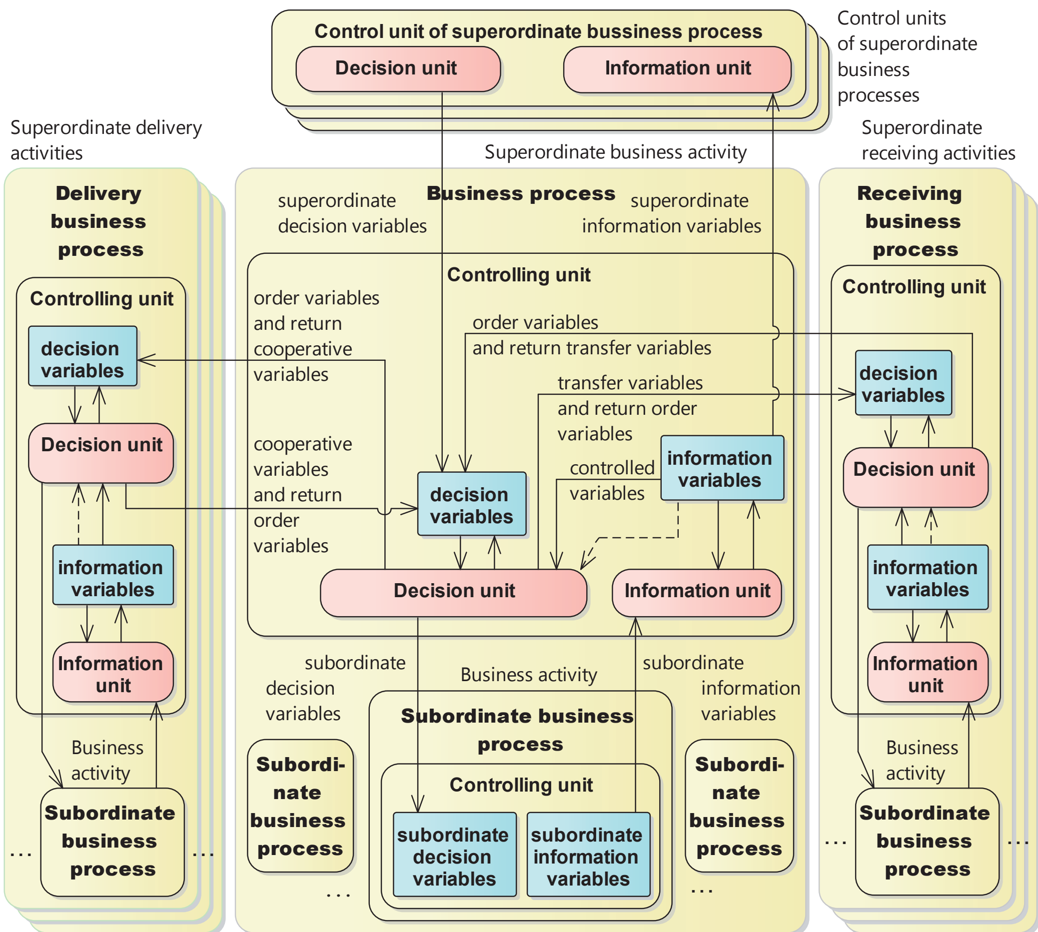

Fig. 6. Information flow between functional units of a business process

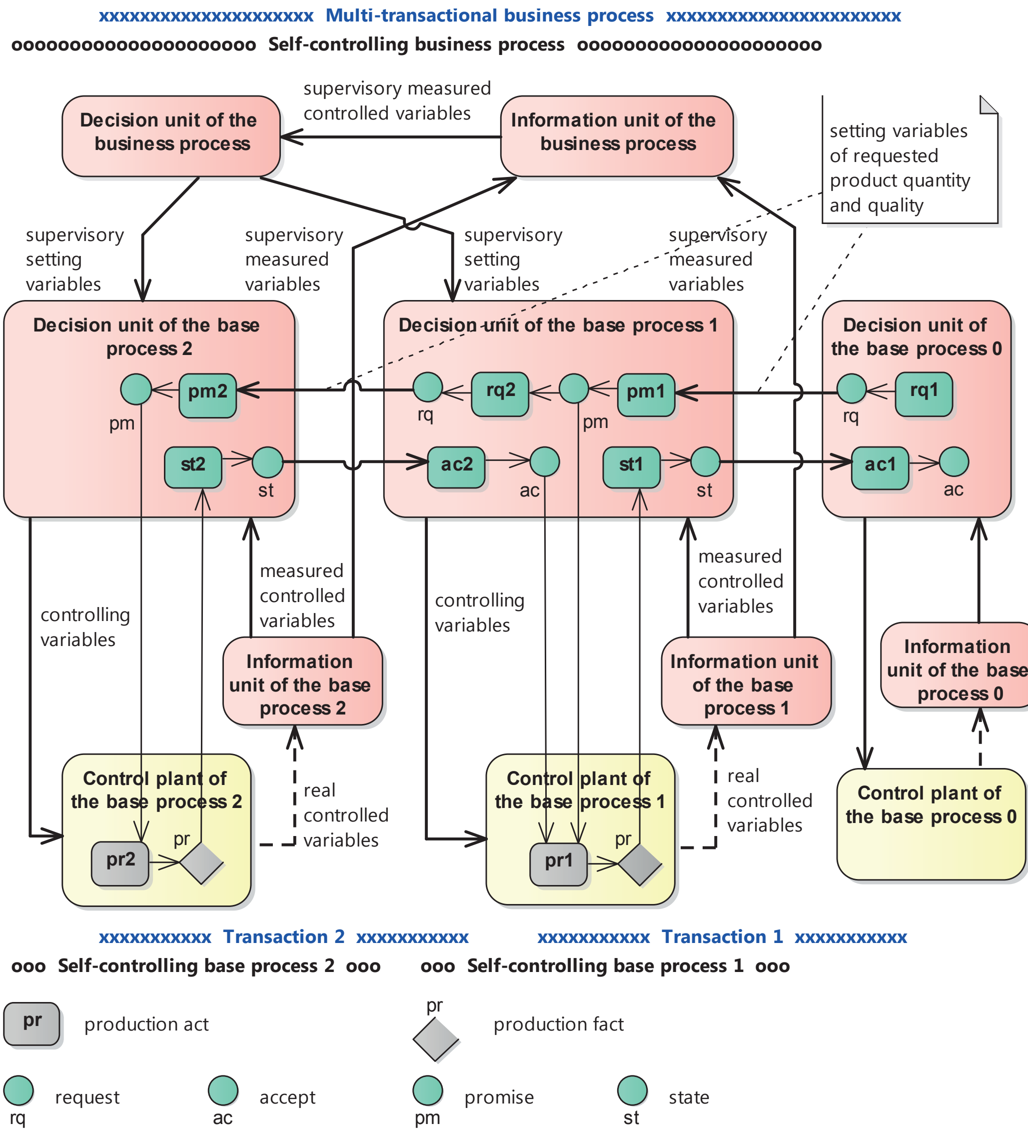

Fig. 7. Acts and facts that belong to the illustrative multi-transactional business process as elements of a corresponding self-controlling enterprise processes

In a general case, a given business process may have many superordinate processes (including business systems and their roles, regarded as enterprise processes), many subordinate activities, many receiving processes and many delivery processes. Thus, the controlling unit of a business process (Fig. 6) may have couplings with many superordinate controlling units, with many controlling units of subordinate activities and with many controlling units of delivery and receiving processes.

An extensive and thorough description of transactional couplings between actors participating in business processes is a part of the PSI theory (Dietz, 2006b). It defines a transaction as a sequence of acts that belong to two actor roles, initiator and executor (Fig. 7), and a business process as "a process that consists of an ordered collection of transaction types". In a general case, a realisation of a multi-transactional business process is "a tree structure of enclosed transactions. A transaction T2 is enclosed in transaction T1 if T2 is initiated by the executor of T1" (Dietz, 2006a).

In the PSI theory, internal structure and the functions of production acts "belong to the realm of implementation" (Dietz, 2006b). In the EntPC theory, the internal structure of processes is unknown only for the infrastructure. Therefore, in Fig. 7, transactions are collated with the base processes, although the DEMO methodology (Dietz, 2006a,b), which is based on the PSI theory, is designed for applications at higher organisational levels.

In the transactional model of business processes (Dietz, 2006a,b) the facts that belong to the basic transaction pattern are as follow (Fig. 7):

rq – an order sent by a customer,
pm – an order realisation promised and its execution started by a producer,
st – an order realisation stated by a producer and
ac – an order product accepted by a customer.

In the i-d model of enterprise processes, the following variables correspond to these processes:

- order variables,
- subordinate decision variables,
- controlled variables carrying information on finished production orders, and
- decision variables of receiving processes, concerning withdrawing products made available by a given business process.

In the i-d model, controlled variables (information on finished orders) are not sent directly to a receiver, as in the model of a single transaction (Dietz, 2006a), but they are transformed into transfer decision variables that carry information on products available for withdrawal. This is necessary in the case of products that may be allowed for different receivers. Analogously, cooperative variables carry information on delivery products made available by suppliers.

Apart from basic patterns, there are also standard and cancellation patterns for transactions (Dietz, 2006b). The standard transaction pattern includes facts that belong to the basic pattern and the following additional facts:

dc decline – order declined by a producer,
qt quit – order repetition quit by a customer,
rj reject – product rejected by a customer, and
sp stop – delivery attempts stopped by a producer.

In the i-d model (Fig. 6), they are represented by

- return order variables,
- decision variables of a receiving process,
- return transfer variables,
- transfer variables.

There are many differences between the structures of the self-controlling and multi-transactional processes. First, the stages of self-controlling processes are enterprise activities performed by single business units, whereas in the case of multi-transactional processes each transaction is a sequence of acts that belong to two actor roles. Second, each transaction is controlled only by its initiator, whereas a self-controlling sub-process (enterprise activity) may be influenced not only by many receiving activities but also by many superordinate enterprise processes (Figs. 6 and 7). Third, in EntPC systems, acts and facts resulting from them are separated. What is more, C-acts and corresponding C-facts (e.g. decision events and records of decision i-d state variables) may belong to different actors (Fig. 6). However, despite these differences, all the facts analysed above concerning the multi-transactional model of business processes have their counterparts in the i-d model of enterprise processes.

3. Conceptual model of self-controlling enterprise processes

3.1. The EntPCL metamodel

Each EntPCL object, as an object of an EntPC system software, includes its own set of attributes, its own set of operations that are executed on the attributes and its own set of relationships with other objects. On the other hand, each EntPCL object is an element of the set of all EntPCL objects in the concrete EntPC system, $o \in O$. The EntPCL metamodel, like ArchiMate's metamodel (Iacob et al., 2012), is a set of class diagrams that impose definite relationships between EntPCL objects. A class of EntPCL objects is, like in UML, a generalisation of a set of EntPCL objects that have the same attributes and operations and the same relationships with objects of other classes (Booch, Rumbaugh, & Jacobson, 1999).

Each class of EntPCL objects corresponds to one of the concepts of the EntPC theory. The relationships between these concepts are visualised in class diagrams of the EntPCL metamodel as relationships between corresponding classes. Thus, the EntPCL metamodel may be regarded as a conceptual model (Snoeck, 2014) of EntPC systems. EntPCL, like ArchiMate (Iacob et al., 2012) and UEML (Vernadat, 2002), may be used to model enterprise reference architecture. Diagrams constructed using each of these languages for concrete enterprises present the relationships between objects of their architecture, but the metamodels that determine areas of modelled facts, are different.

The EntPCL metamodel is identical for every enterprise. However, in a specific EntPC system, its classes represent finite sets of objects, whereas the relationships between the classes represent sets of relationships between the objects that belong to the sets. The names of sets of objects that are presented for every object in the EntPCL object diagrams are visible in the class diagrams as short names of corresponding classes. To improve readability of metamodel class diagrams, pictures of classes include also long names of the classes as well as symbols of object identification numbers. The illustrative object diagrams (Figs. 2 and 4) correspond to a fragment of the class diagram shown in Fig. 8.

Aggregation relationships between activities and processes (Fig. 2) correspond to the relation of weak aggregation between classes *B* and *P* of generalised business activities and processes. Order relationships between activities are represented by order relations between class *Bd* of generalised delivery activities and class *B*, as well as between class *B* and class *Bc* of generalised receiving activities. Composition relationships between business activities and business units, as well as between business units and business systems (Fig. 4), correspond to the relations of composition between class *A* and class *U*, as well as between class *U* and class *S*.

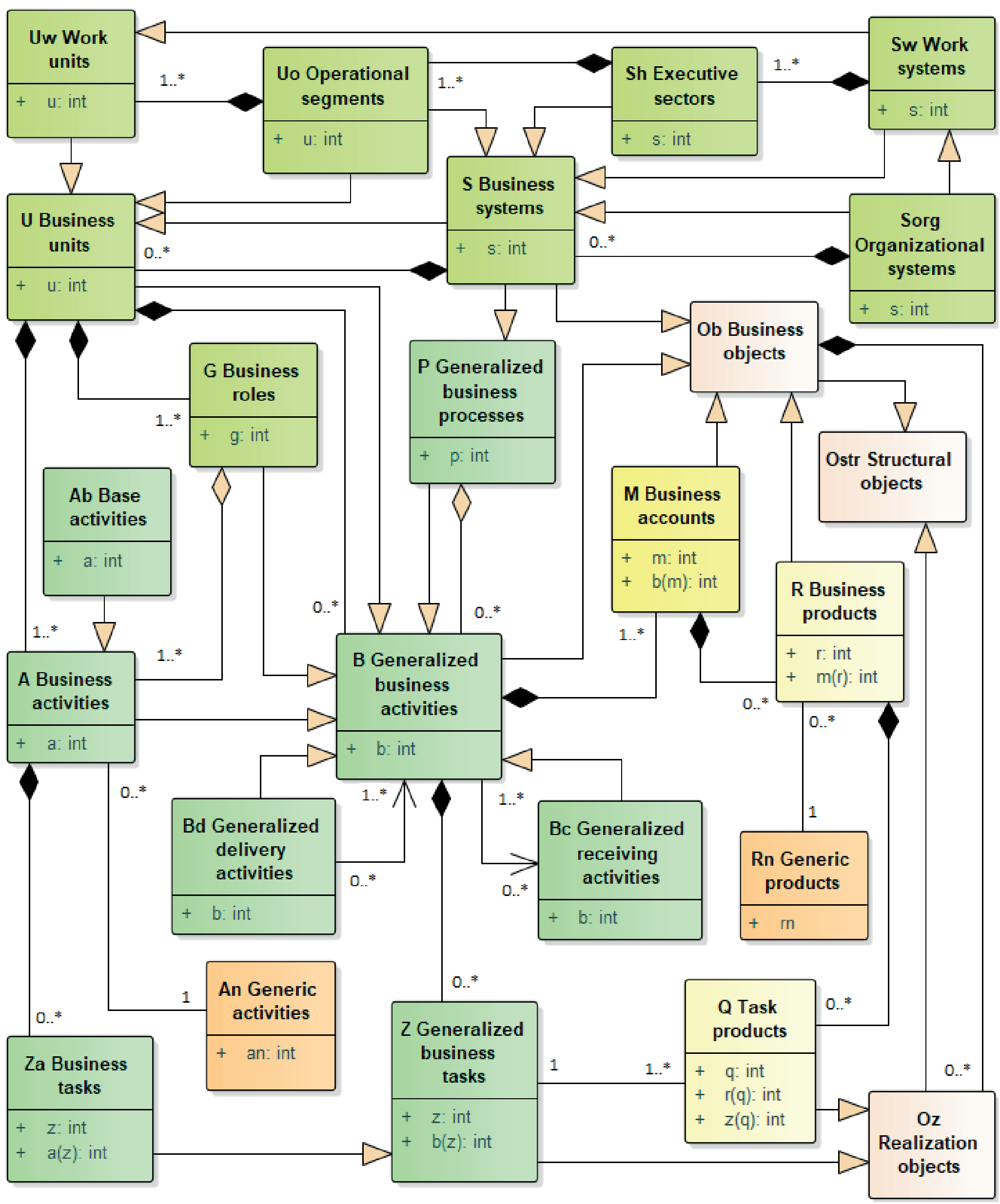

Fig. 8. Class diagram for the relationships between business and realisation objects

Delivery and receiving business activities, as well as business processes and business units, are special cases of generalised business activities, but this is not visible in the object diagrams. This is shown correspondingly in the class diagram as generalisation relations between classes *Bd, Bc, P, U* and class *B* (Fig. 8). There is also a compositional relation between classes *S* and *U*.

3.2. Structural objects

The EntPCL metamodel encompasses all structural objects of EntPC systems and all variables that are their attributes, as well as all associations that represent their relationships. Structural objects

$$Ostr = Ob \cup Oz$$

are divided into business objects

$$o \in Ob \supset B \cup M \cup R,$$

and realisation objects (Fig. 8),

$$o \in Oz \supset Z \cup Q.$$

Business activities, $a \in A$, business processes, $p \in Pa$, business tasks, $z \in Za$, business works, $w \in Wa$, and generalised business activities, $b \in B$, have been defined in the chapter 1. Generalised business tasks,

$$z \in Z = Za \cup Zu \cup Zg,$$

include not only business tasks, but also collected business tasks, $z \in Zu$, and group business tasks, $z \in Zg$, which are group executions of all business activities that belong to definite business units, $u \in U$, or to definite business roles, $g \in G$.

Business accounts, $m \in M$, that belong to a definite generalised business activity are places of information on this activity and its executions. Formally, they may be presented as components of generalised business activities, which may be divided into accounts of business activities, accounts of business units and accounts of business roles:

$$M = Ma \cup Mu \cup Mg$$

Each generalized business activity has at least three business accounts, belonging to the three corresponding subclasses:

$$m \in M \supset Min \cup Mout \cup Mb$$

Two of them group information on input and output products of a given activity. The third is a place of information related directly to the activity.

A business product, $r \in R$, is a generic product, $r^n(r) \in Rn$, that is produced or used in a definite generalised business activity and belongs to a definite account, $m(r) \in M$, of this activity, $b(m(r)) \in B$. Task products, $q \in Q$, are business products that are attributed to concrete generalised business tasks. Tasks products are components of business products, which are components of business accounts and business accounts are components of generalised business activities, which in turn are components of business units and organisational systems (Fig. 8). Thus, all structural objects belong to a structure tree (similar to the one shown in Fig. 4), whose root is the enterprise as a whole (Zaborowski, 2016b).

3.3. Functional variables and information-decision state variables

The changeable attributes of business objects and realisation objects are called respectively business variables, $i \in Ib$, and realisation variables, $i \in Iz$, (Fig. 9). Formally, they are components of business objects and realisation objects and, on the other hand, they are attributed to corresponding generic variables, $i \in In$.

Business and realisation variables are attributes of structural objects (Fig. 9). They are processed by the functional units of self-controlling enterprise processes. Therefore, they have been named functional variables,

$$i \in I = Ib \cup Iz.$$

Functional variables are divided into information variables, $i \in Ii \subset I$, and decision variables, $i \in Id \subset I$, which are, respectively, the output variables of information and decision transitions:

$$i \in I = Ii \cup Id,\ Ii \cap Id = \emptyset,$$

The set of functional variables includes the following:

- quality variables, e.g. length, diameter, colour, and temperature;
- time variables, e.g. the due date of a business task;
- existential variables, i.e. binary variables that indicate whether specific business objects exist,
- guard variables, i.e. binary functional variables that are used to control the executions of business transition operations.

In control systems, the variables attributed to a specific moment in time are often referred to as signals (Bubnicki, 2005). Therefore, the values $y_i(l,t)$ of functional variables, $i \in I$, recorded at

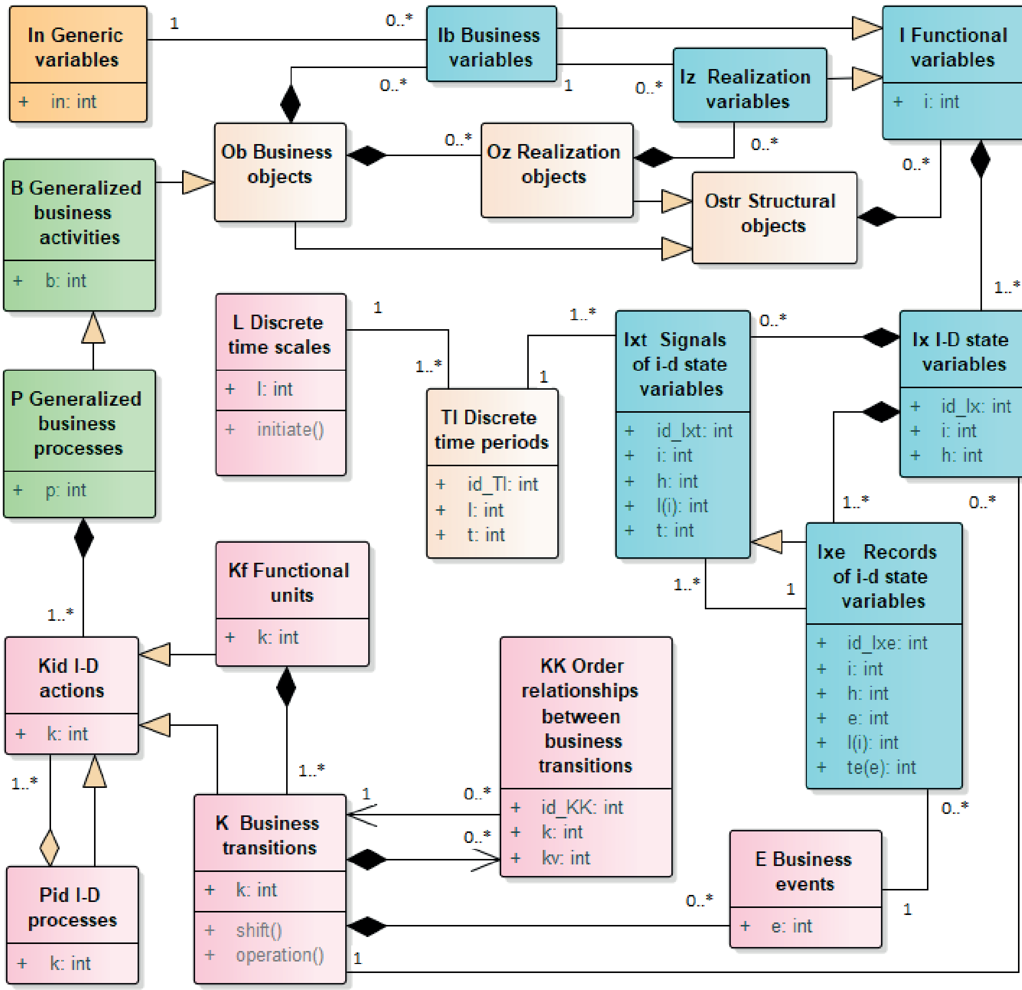

Fig. 9. Discrete time, business transitions and i-d state variables

discrete-time instants, $(l,t) \in Tl$, are values of the signals of functional variables,

$$(i, l, t) \in It \subset I \times L \times T.$$

Knowledge of the current values of functional variables is not sufficient to control enterprise processes. One should also know the values of the i-d state variables that are assigned to the instants $(l,t+h)$, shifted in time, back or forward, by a definite number h of discrete-time periods (see introduction). The value,

$$x_i(l, t, h) = x_{ih}(l, t),$$

of a signal,

$$(i, h, l, t) \in Ixt \subset I \times H \times L \times T,$$

of an i-d state variable,

$$(i, h) \in Ix \subset I \times H,$$

at a specific time instant, is equal to the value of the signal of a functional variable that is shifted in time by a definite number of discrete-time periods, $h \in H$, of the time scale, $l \in L$, applied to this functional variable,

$$x_{ih}(l, t) = y_i(l, t + h), \quad (4)$$
$$for\ h_i^- \le h \le h_i^+,\ i \in I,\ (l, t) \in Tl.$$

Conversely, the value of a functional variable signal is equal to the value of the signal of the i-d state variable with a zero-time shift:

$$y_i(l, t) = x_{ih}(l, t) \mid h = 0 \wedge (i, h) \in Ix, \quad (5)$$
$$for\ t_l^- \le t \le t_l^+,\ l \in L,\ i \in I.$$

Formally, an i-d state variable is a component of a functional variable (Fig. 9) and, indirectly, a component of a specific business object, a specific business activity and a specific business unit. One functional variable may correspond to many i-d state variables. I-d state variables, like functional variables, are divided into information state variables, *(i,h)* $\in$ *Ixi* $\subset$ *Ix,* and decision state variables, *(i,h) Ixd* $\subset$ *Ix.*

Moving to a new discrete-time instant does not change the values of i-d state variables but does change their identifiers. Therefore, immediately after creation (by a clock), the initial instant of a current discrete-time period, *(l,t)* $\in$ *Tl,* for a given time scale, *l* $\in$ *L,*

$$(l,t) \coloneqq (l,t+1), \quad (6)$$

and prior to making current information on the i-d state, one should decrease the values of the time shifts of the i-d state variables by 1 relative to the current time instant (Zaborowski, 2018).

Business events that are regarded as executions of business transitions may insert the records of i-d state variables,

$$(i,h,e) \in Ixe \subset Ix \times E,$$

into the system memory. Each record *(i,h,e)* of an i-d state variable *(i,h)* is also an effect of one definite event, *e* $\in$ *E,* and is a formal component of this variable. Each record of an i-d state variable corresponds to the i-d state variable signal at the instant the record is created and, perhaps, to the signals at certain future time instants.

Access of business transitions to their input i-d state variables and general description of procedures for processing i-d state variables are discussed in (Zaborowski, 2018).

Conclusions

Self-controlling enterprise processes have been defined in this study as a new category of business processes. It encompasses not only production, preparatory and managerial business processes, but also business systems and their roles. Events, gateways, conditions and other elements that are used for managing the sequence of task executions have been grouped in one controlling unit of a given process. Additionally, all business transitions, which control the quantity and quality of products that are processed in a production-oriented business process, are also included in its controlling unit. It has been demonstrated that integrated management and direct control systems may be built as integrated enterprise process control systems (EntPC systems), which are networks composed of infrastructural control plants and of controlling units of multistage, multilevel self-controlling enterprise processes.

To model the structures of concrete business processes and the details of the structures of process management systems, one can use EntPCL diagrams, which are UML object diagrams that fulfil the structural constraints imposed by the class diagrams of the EntPCL metamodel. The EntPCL metamodel is a conceptual model of EntPC systems. It is the base of the Enterprise Process Control Framework (EntPCF), which is a description of the structure and behaviour of EntPC systems, and it underlies the Enterprise Process Control (EntPC) theory. The most important of many EntPCL class diagrams have been presented in this study. They describe relationships between subclasses of structural objects and between structural objects and information-decision state variables of EntPC systems.

Obviously, the EntPCF may be used as a framework for comparing different management methods and algorithms, e.g. IT tools for Business Intelligence development (Olszak & Żurada, 2015). In addition, the i-d state variables of the controlling system, together with the state variables of infrastructural processes, are state variables for any entire EntPC system. Therefore, the EntPC theory may facilitate transferring the results of the classical control theory to the systems of enterprise management, e.g., to analyse enterprise stability and controllability. It is especially useful for the industry 4.0 enterprises (Kagermann et al., 2013), because their management systems should react in real time to the enterprise state changes (Youssef et al., 2017) and, on the other hand, real-time control systems are the subject of the control theory.

The i-d model of enterprise processes differs essentially from the currently dominant standards for modelling business processes. Moreover, a significant number of new concepts and new interpretations of concepts that belong to three different domains — cybernetics, informatics and management science — may discourage the interest in the EntPCF. Therefore, in addition to the cognitive values discussed previously, one should show the practical benefits that could result from its application. It is also important to present examples of modules of ERP, MES, SCADA and PLC systems that may be implemented as corresponding modules of EntPC systems. A method of embedding such modules in the EntPCF structure has been presented (Zaborowski, 2016a) by the example of the well-known MRP algorithm (Orlicky, 1975).

In practice, it is important that class diagrams of the EntPCL metamodel are similar to class diagrams of the UML, which is designed for modelling software of IT systems. Therefore, it may be the starting point for creating the software framework for enterprise process control (SFEntPC), which will be used for designing executable models and generating software for concrete EntPC systems. The tree structure of composition relationships between organisational systems, enterprise processes, business transitions, structural objects and i-d state variable (Figs. 8, 9) should facilitate implementation of the SFEntPC as an extension to the Eclipse Modeling Framework (EMF) (Steinberg et al., 2008).

The controlling units of enterprise processes (Fig. 1) will be replaceable building blocks in the software generated in the SFEntPC environment. Thus, re-engineering of a given self-controlling enterprise process, perceived as a branch in an enterprise structure tree, relies on removing from it the controlling units and infrastructural processes that belong only to the sub-processes selected for elimination and embedding in it the complete controlling units and infrastructural processes of new sub-processes. The enterprise itself is a self-controlling enterprise process as well. So, this type of re-engineering of enterprise processes is also re-engineering of the enterprise software. Such an operation may be performed by business analysts, without the participation of IT engineers. This will obliterate the "business-IT divide", which refers to the necessity of difficult and prolonged arrangements between business analysts, who understand the actual goals of process re-engineering, and IT engineers, who are authorised to make changes to the structure of management systems software (Smith & Fingar, 2003).

Acknowledgements

This work was supported by the WSB University.

Literature

van der Aalst, W., & van Hee, K. M. (2002). *Workflow Management: Models, Methods, and Systems Cooperative Information Systems Series.* Cambridge, United Kingdom: MIT Press.

Badura, D. (2014). Modelling business processes in logistics with the use diagrams BPMN and UML. *Forum Scientiae Oeconomia, 2*(4), 35-50.

Bernus, P., Noran, O., & Molina, A. (2015). Enterprise architecture: Twenty years of the GERAM framework. *Annual Reviews in Control, 39,* 83-93. doi: 10.1016/j.arcontrol.2015.03.008.

Blackstone, J. H., & Cox, J. F. (2005). *APICS Dictionary.* 11th ed. Falls Church, VA: American Production and Inventory Control Society.

Booch, G., Rumbaugh, J., & Jacobson, I. (1999). *The Unified Modeling Language. User Guide.* Boston, United States: Addison-Wesley.

BPMN. (2013). Business Process Model and Notation, Ver.2.0.2, OMG.

Bubnicki, Z. (2005). *Modern Control Theory.* Berlin, Germany: Springer-Verlag.

Davis, R., & Brabander. E. (2007). *Aris Design Platform: Getting Started with BPM.* Berlin, Germany: Springer-Verlag.

Dietz, J. L. G. (2006a). The Deep Structure of Business Processes. *Communications of ACM, 49*(5), 58-64. doi: 10.1145/1125944.1125976.

Dietz, J. L. G. (2006b). *Enterprise Ontology: Theory and Methodology.* Berlin, Germany: Springer-Verlag.

Hofstede, A. H. M., van der Aalst, W., Adams M., & Russell, N. (2010). *Modern Business Process Automation: YAWL and Its Support Environment.* Berlin, Germany: Springer.

Iacob, M., Jonkers, H., Lankhorst, M., Proper, E., & Quartel, D.A. (2012). *Archimate 2.0 Specification.* The Open Group.

Kagermann, H., Wahlster, W., & Helbig, J. (2013). Recommendations for implementing the strategic initiative Industrie 4.0, *Final report of the Industrie 4.0 Working Group,* Acatec.

Kosanke, K., Vernadat, F., & Zelm, M. (1999). CIMOSA: enterprise engineering and integration. *Computers in Industry, 40*(2-3), 83-97.

Langenwalter, G. A. (2000). *Enterprise Resources Planning and Beyond: Integrating Your Entire Organization.* New York, United States: St. Lucie Press.

Lockemann, P. C. (2006). Agents. *Multiagent Engineering: Theory and Applications in Enterprises.* S. Kirn, O. Herzog, P. Lockemann, & O. Spaniol (Eds.) (pp. 17-34). Berlin, Germany: Springer-Verlag

Mesarović, M. D., Macko, D., & Takahara, Y. (1970). T*heory of Hierarchical, Multilevel, Systems.* New York, United States: Academic Press.

Monostori, L., Valckenaers, P., Dolgui, A., Panetto, H., Brdys, M., & Csáji, B. C. (2015). Cooperative Control in Production and Logistics. *Annual Reviews in Control, 39,* 12-29. doi: 10.1016/j.arcontrol.2015.03.001.ąq\aQq

Murill, P. W. (2000). *Fundamentals of Process Control Theory* (3rd Edition). ISA Publ.

Noran, O. (2003). An analysis of the Zachman framework for enterprise architecture from the GERAM perspective. *Annual Reviews in Control, 27,* 163-183.

Olszak, C. M., & Żurada, J. (2015). Information technology tools for Business Intelligence development in organizations. *Polish Journal of Management Studies, 12*(1), 132-142.

Orlicky, J. (1975). *Material Requirements Planning.* New York, United States: McGraw-Hill.

Panetto, H. (2007). Towards a classification framework for interoperability of enterprise applications. *International Journal of Computer Integrated Manufacturing, 20*(8), 27–740. doi: 10.1080/09511920600996419

Reijers, H. (2003). *Design and Control of Workflow Processes, 2617*. Berlin, Germany: Springer-Verlag.

Saha, P. Analyzing TOGAF from the GERAM perspective. Retrieved from http://www.opengroup.org/architecture/wp /saha/TOGAF_GERAM_Mapping.htm

Scholten, B. (2007). *The road to integration. A guide to applying the ISA-95 standard in manufacturing.* Research Triangle Park, NC: ISA Publ.

Smith, H., & Fingar, P. (2003). *Business Process Management: The Third Wave.* Tampa, FL: Meghan-Kiffer Press.

Snoeck, M. (2014). *Enterprise Information Systems Engineering. The MERODE Approach.* Springer.

Steinberg, D., Budinsky, F., Paternostro, M., & Merks, E. (2008), *EMF: Eclipse Modeling Framework*. Addison-Wesley.

Vernadat, F. (2002). UEML: Towards a unified enterprise modeling language. *International Journal of Production Research, 40*(17), 4309-4321. doi: 10.1080/00207540210159626

Weske, M. (2012). *Business Process Management: Concepts, Languages, Architectures*. Berlin, Germany: Springer-Verlag.

WFMC, (1999). Terminology and Glossary, Doc. N. Wfmc-Tc-1011, Issue 3.0.

Williams, T. (1994). The Purdue Enterprise Reference Architecture. *Computers in Industry, 24*(2-3), 141-158.

Youssef, J. R., Zacharewicz, G., Chen, D., & Vernadat, F. (2017). EOS: enterprise operating systems. *International Journal of Production Research, 55*.

Zaborowski, M. (2009). Outline of the Enterprise Resource-Control Systems Architecture. In E. Tkacz & A. Kapczynski (Eds.), *Internet – Technical Development and Applications. Advances in Intelligent and Soft Computing, 64*. Berlin, Heidelberg, Germany: Springer.

Zaborowski, M. (2011). The EPC II theory. Data Structure in Enterprise Process Control Systems. *Management and Production Engineering Review, 2*(2), 53-70.

Zaborowski M. (2015). Tabular Models of Business Process Structures. In M. Rostański, P. Pikiewicz & P. Buchwald (Eds.), *Internet in the Information Society* (pp. 281-298). Dabrowa Gornicza, Poland: Publ. University of Dabrowa Gornicza.

Zaborowski, M. (2016a). *Introduction to the Theory of Enterprise Process Control.* Dabrowa Gornicza, Poland: Publ. University of Dabrowa Gornicza.

Zaborowski, M. (2016b). Generalization and Composition-Relationships between Objects of Enterprise Process Control Systems. In M. Rostański, P. Pikiewicz, P. Buchwald & K. Maczka (Eds.), *Proceedings of the 11th Scientific Conference Internet in the Information Society 2016* (pp. 405-418). Dabrowa Gornicza, Poland: Publ. University of Dabrowa Gornicza.

Zaborowski, M. (2018). Data processing in self-controlling enterprise processes. Working paper. Retrieved from https://www.researchgate.net/profile/Miroslaw_Zaborowski2

Appendix

Symbol	Description
A	business activities
Ab	base business activities
An	generic activities
B	generalised business activities, enterprise activities
Bc	generalised receiving activities
Bd	generalised delivery activities
E	business events
G	business roles
I	functional variables
Ib	business variables
Id	decision variables
Ii	information variables
In	generic variables
It	signals of functional variables
Ix	i-d state variables
Ixd	decision state variables
Ixe	records of i-d state variables
Ixi	information state variables
Ixt	signals of i-d state variables
Iz	realisation variables
K	business transitions
Kd	decision transitions
Kf	functional units
Ki	information transitions
Kid	information-decision actions
L	discrete-time scales and their clocks
M	business accounts
Ma	accounts of business activities
Mb	accounts directly related to activities
Mg	accounts of business roles
Min	input business accounts
$Mout$	output business accounts
Mu	accounts of business units
$Ostr$	structural objects
Ob	business objects
Oz	realisation objects
P	generalised business processes, enterprise processes

Pa	business processes
Pb	base processes
Pg	business system roles, group processes
Pid	information-decision processes
Q	task products
R	business products
Rin	input products
Rn	generic products
$Rout$	output products
S	business systems
Sh	executive sectors
$Sorg$	organisational systems
Sw	work systems
Tl	discrete-time periods
U	business units
Uh	executive units