MODELLING BUSINESS PROCESSES IN LOGISTICS WITH THE USE OF DIAGRAMS BPMN AND UML

Dariusz Badura, Prof.

drhbad@gmail.com

University of Dąbrowa Górnicza, Poland

Abstract

The paper researches the suitability and sufficiency of BPMN and UML languages in business process modelling in logistic. There were two examples of process modelling: supply chain logistics and production logistics. Logistics system is modelled at the highest domain level using case study in container transport and exemplary production process to examine the both modelling languages in many aspects. Modelling is carried out in a modular basis for hierarchical systems generally using BPMN 2.0 and UML 2.0. Complementary use of both modelling tools was seen necessary for business process flow and information process design. The proposed models provides many benefits for modellers, analytics and actual practitioners of logistics activities.

Keywords: BPMN diagram, UML activity diagram. business process modelling, logistic process, information system

Introduction

Business process modelling is the important element in Business Process Management at their implementation phase as well as the activity of representing the processes of an enterprise. The experiments in the real world are usually very expensive and do not allow to take into account full number of parameters these processes. Business Process Management (BPM) requires a specific description language, the graphical representation of such a description is the best way to communicative presentation of the proposed solution.

Process modelling becomes more and more an important task not only for the purpose of software engineering but also for many other purposes besides the development of software. Therefore it is necessary to evaluate

the quality of process models from different viewpoints. This is even more important as the increasing number of different end users, different purposes and the availability of different modelling techniques and modelling tools leads to a higher complexity of information models.

In the information system design process there are used two approaches:

- Object oriented approach, in which there are identified objects and used such facilities as abstraction, encapsulation, inheritance and polymorphism.
- Functional oriented approach, in which there are identified functions and activities of system. A group of primitive activities can be assigned to a particular role.

In the actual design process, both approaches are used simultaneously with a focus on one of them. In the object-oriented approach, objects are identified, and for them, in turn, are defined responsibility implemented using the identified functions (or methods). Such an approach in modelling business processes can be incomprehensible for the business side. A good support for business process modelling is activity-oriented approach. It tend to define a business process as a specific ordering of activities (i.e. tasks).

At the present, there are several modelling languages which define the basic elements for constructing business process models. The mains are in historical order:

- Petri nets and colour Petri nets (Petri 1962).
- Event-driven Process Chains (EPC) used within ARIS framework (Aalst 1999).
- UML Activity Diagrams, the process modelling language of UML (OMG 2007).
- Yet Another Workflow Language (YAWL) is built on the workflow pattern analysis (Aalst, Hofstede 2002; Aalst et al. 2003).
- Business Process Execution Language for Web Services (BPEL4WS) to model executable processes (OASIS 2007).
- Business Process Modelling Notation (BPMN) to describe the business processes using notations standardized by OMG (OMG 2011).
- Business Process Modelling Language (BPML) is a language for business process modelling. BPML was a proposed language, but now the BPMI (Business Process Management Initiative) has dropped support for this in favour of BPEL4WS (BPML 2002).

The considerations made in this article will be focused on two notations: UML activity diagram and BPMN. To effectively describe business processes, the UML language undergone a significant evolution from simple graphical language describing software project to very complex tool to describe the various elements of business processes. However, UML supports

primarily object-oriented approach. The most important UML diagrams include the use case diagram, activity diagram, state diagram and class diagram. From the point of view of business process modelling activity diagram is the most interesting. Table 1 presents a comparison of the activity diagram elements with elements of BPMN diagram.

Business Process Modelling Language as a metalanguage for modelling business processes and business data provides an abstracted execution model for collaborative and transactional business processes based on the transactional finite-state machine concept. The UML takes an object-oriented approach to the modelling of applications, whereas BPMN takes a process-oriented approach to modelling of systems. The BPMN has a focus on business processes description, the UML has a focus on software design modelling. Thus, both notations are not competitive, but are different views on systems. The BPMN and the UML are compatible with each other. A business process model does not necessarily have to be implemented as an automated business process in a process execution language. It is a way for the exchange of knowledge between engineers and representatives of various fields. There is the case, in which business processes and participants can be mapped to constructs such as use cases and behavioural models in the UML.

BPMN is a standard set of graphical diagramming conventions for describing business processes. It allows to visualize a rich set of process flow semantics within a process and the communication between independent processes. It has been designed to support capture of sufficient detail to allow it to be the base of an executable process description. There are several languages aimed coding and implementation of the models described BPMN diagrams. Among others these are BPEL and BPML.

BPEL is language based on XML standard. It was design for describing a business process in which most of the tasks represent interactions between the process and external Web services. The BPEL processes themselves are represented as a Web services, and are realized by a BPEL engine which executes the process description. A translation to BPEL is specified in the BPMN standard, because BPEL is currently considered the most important standard for execution languages. But by design there are some limitations on the process topologies that can be described in BPEL, so it is possible to represent processes in BPMN that cannot be mapped to BPEL.

BPML was developed by the Business Process Management Initiative to model business processes and it is also based on XML standard. It has been dropped in support of Business Process Execution Language. Activities in a BPML perform specific functions and are either simple or complex. Simple activities such as action, assign, call, compensate, etc., cannot be further decomposed and perform a single operation. Complex activities such as all,

sequence, flow, switch, pick, etc., are composed of one or more activities and direct the execution of an activity from another activity set. BPML and BPEL are intended for mission-critical applications by supporting synchronous and asynchronous distributed transactions.

Why BPMN has become so popular? Users expect notation, which will allow modelling of advanced composite business processes. A proposed notation emerges not to solve not only old problems. It should be competitive in traditional applications and open new opportunities. A model can be analyzed and improved organization's processes.

BPMN provides a notation that can be readily understandable by all users: from the business analysts who model the processes conceptually, to the technical developers responsible for implementing the technology for the processes, to the people who will manager and monitor the processes. Both the business and technical sides of the organization can find a common language. They can both understand and that meets their respective needs for precision and notation flexibility. This shared language give the opportunity to use new ways of working together. It results in the deployment of new and more flexible applications.

1. BPMN and UML Activity Diagram notation

BPMN generally defines an abstract model and the grammar used to express a generic process. As such, it can be used to define enterprise business processes, complex Web services and multiparty collaborations.

The BPMN can approach by organizing elements in four categories:

- Workflow elements:
  - activities,
  - events,
  - gateways,
  - sequence flow,
- Organizing elements:
  - pools,
  - swimlanes,
  - groups,
- Readability elements:
  - annotation,
  - links,
- Special behaviour elements:
  - messages,
  - message flow,
  - signals,
  - timers,

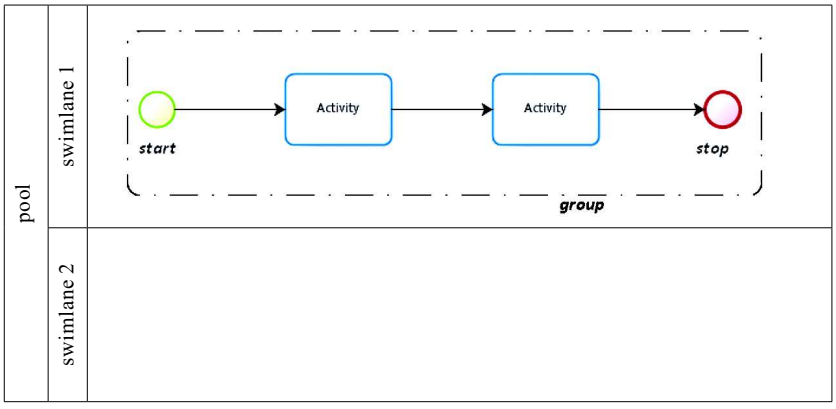
– errors,
– repeating,
– correlation.

With just a few elements from first three categories there can draw a business process diagram and begin to build and understand a process. Each of Workflow elements have several types, and all of these types can be connected in a sequence. Activities means tasks that are performed in the process by humans, by automation, or by subprocesses. Events are used to start or end process, and also to manage specific actions during a workflow. There are three type of events: start, intermediate and end events. *Intermediate events* which will affect the flow of the process, occur between a *start event* and an *end event*. They will not start or directly terminate the process. Gateways are used to separate or joint process flow. The described elements can be linked together by an arrow representing a sequence of flow (Figure 1)

Figure 1. Symbols of workflow elements: a) activities, b) events, c) gateways, d) sequence flow

Organizing elements are used to collect of the process flow elements. Pool contains a single, complete process. Workflow cannot leave a pool. A transfer action or data from one pool/process to another can be realized using events. A swimlane is used to help organize the process based on roles i.e. who does what. Workflow crosses swimlane boundaries as if they did not exist. They are purely for organizational clarity. Group is used to enclose a grouping of featured graphical elements. It does not affect sequence flow (Figure 2).

Figure 2. Using of organizing element symbols: pool, swimlanes, groups

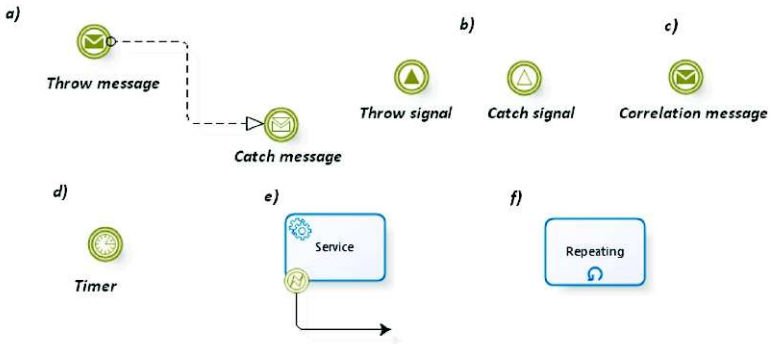


Readability elements have no effect at all on the actual process flow. Text annotation allow to paste notes all over a model with explanations for clarity. Links allow to cut a process that has become too long to read easily, and simply continue the process on another line (Figure 3).

Figure 3. Symbols of readability elements: a) text annotation, b) input and output links

Special behaviour elements allow to design executable workflow that can behave in complex ways. Messages and message flow elements are used to transfer action or data from one pool/process to another and to correlate related processes. Signals are used to send data to multiple activities simultaneously. Correlation is used to coordinate progress between to running process instances. Timers are used to launch periodic activities, or to ensure that an activity happens within a specified deadline. Errors are used to define behaviour when the system encounters error. Repeating is used to repeat behaviour, such as multiple launches of the same task, or repeating the same task multiple times (Figure 4).

Figure 4. Symbols of special behaviour elements: a) transfer action by throw and catch message; b) throw and catch signal element; c) correlation event (message); d) time symbol element; e) symbol of activity with error service; f) symbol of repeating activity

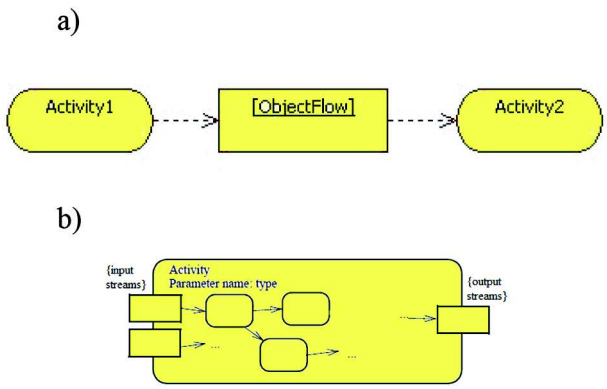


BPMN and UML are built in a hierarchical manner, starting from the general form of the system model, through successive clarification processes or activities, to the detailed model. In the final step of the modeling, processes become elementary activities or tasks. The process, which is not

divided on the simpler processes, can be explained at the next level of the hierarchy. This process (collapsed process) indicates its development by a sign "+" inside the process/activity symbol. Number of hierarchy levels depends on the complexity of the entire process, because the developed process in the next level can also contain collapsed processes.

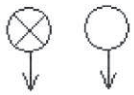
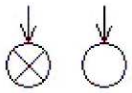
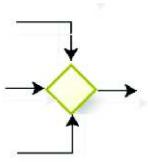
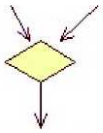
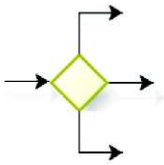
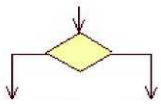
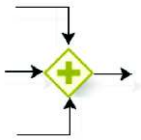
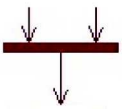
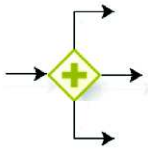
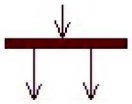
BPMN and UML activity diagrams have similar features and ease of use. Both BPMN and UML AD allow the modeling of complex sequences of tasks and activities. But UML supports primarily object-oriented approach. In addition to activity nodes and control nodes may appear nodes objects. Typically object node connects two nodes of activity (Figure 5a). This means that the object presented is the result of the activity and it is passed on to the next activity. Symbol of the activity node can contain rectangles indicate input and output parameters and streams. Thus, the UML activity diagram describes the flow of data and information processing in the modelled system.

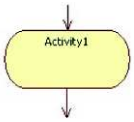
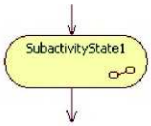
Figure 5. a) Two object flow arrows linking object nodes and actions; b) Notation of the activity with input and output streams



Although the objectives of the use of diagrams, BPMN and UML AD are different elements of the two types of diagrams are similar, they have similar meanings. Table 1 shows the comparison of major elements of the two graphical languages. The table does not include the full set of events and the types gates. They are not exact equivalents in the activity diagram for the BPMN elements. Also, do not treat the presented elements and their interpretation as a complete set of graphical symbols and their meanings. The complete description can be found in the standards of both languages BPMN and UML announced by the OMG (Object Management Group) (OMG 2011; BPML 2002).

Table 1. Comparison of BPMN and UML AD diagrams elements

BPMN	Description	UML activity diagram	Description
	Start Event indicates where a particular Process or Choreography will start		Control nodes: Beginning of Activity Flow
	End Event indicates where a Process or choreography will end		Control nodes: End of Activity Flow
	Off-Page Connector will show where a Sequence Flow		Control nodes: Activity edge connector – input
	leaves one page and then restarts on the next page.		Control nodes: Activity edge connector – output (flow final)
	Merging: exclusive combining of two or more paths into one path (OR-Join). A Merging Exclusive Gateway is used to show the merging of multiple Sequence Flows		Control nodes: Merge node
	Fork: Multiple Outgoing Sequence Flows This will be used usually in combination with any Gateway.		Decision node with control flows
	Gateway Control Type Parallel Gateway joining		Control nodes: Join node
	Gateway Control Type Parallel Gateway forking		Control nodes: Fork node with flows

	Gateway Control Type: inclusive, event- based, complex	---	
	Intermediate events: "Throwing" signal and message		Send Signal Activity Action
	Intermediate events: "Catching" signal and message		Receive Signal Activity Action
	Timer intermediate event		Action of accept time event generating
	An Activity is a generic term for work that company in a Process. An Activity can be atomic or compound.		Action from Complete Activities, Fundamental Activities, Structured Activities, Complete Structured Activities
	The types of Activities that are a part of a Process Model are: Sub- Process and Task. A Sub-Process is a compound Activity that is included within a Process. It is compound in that it can be broken down into a finer level of detail through a set of sub-Activities.		Sub-Activity

2. Exemplary logistic process model

In business, logistics may have either an internal focus (inbound logistics) or an external focus (outbound logistics), covering the flow and storage of materials from point of origin to point of consumption (supply-chain management). The main functions of a qualified logistician include inventory management, purchasing, transportation, warehousing, material handling, packaging, security, consultation, and the organizing and planning of these activities. Responsive logistics system all along the supply chain is severely dealing with workflow management system. Logisticians combine a professional knowledge of each of these functions to coordinate resources in an organization.

Examples of the use of BPMN diagrams can be found in the literature (Khabbazi et al. 2013a; Khabbazi et al. 2013b; Koniewski et al. 2006).

The article (Koniewski et al. 2006) describes a simple model of logistic chains in which additionally Petri nets were used to simulate the processes. In this section Business Process Model and Notation is used to represent the functional structure and activities of reference logistics system applicable in small and medium enterprises (SME) environment. Below are examples of modelling the two processes: Multimodal Logistics Chains and Production Logistics.

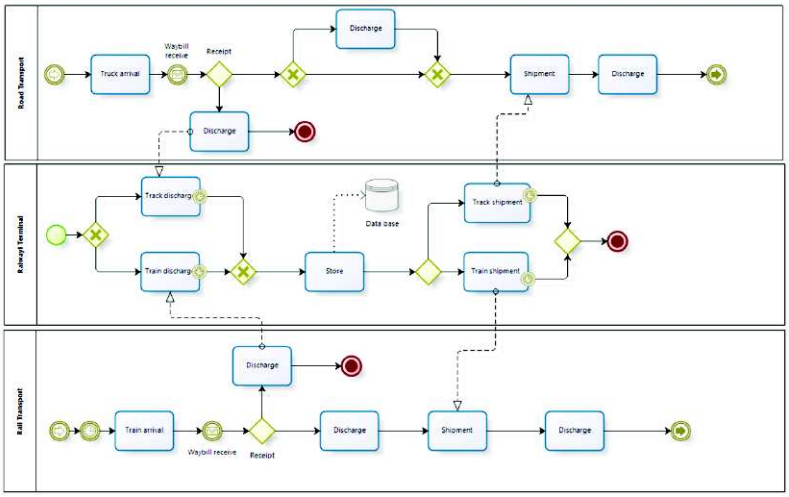
2.1. Modelling of multimodal logistics chains

Multimodal logistic chain is the interconnected set of business procedures and business partners that manage the flow of goods and information from the one point e.g. production, through intermediate nodes, to the another one e.g. to the end consumer. In this example will be presented a problem, in which the transportation unit is a single container. In such a situation we can distinguish three basic types of nodes:

- road terminal,
- train terminal,
- sea port.

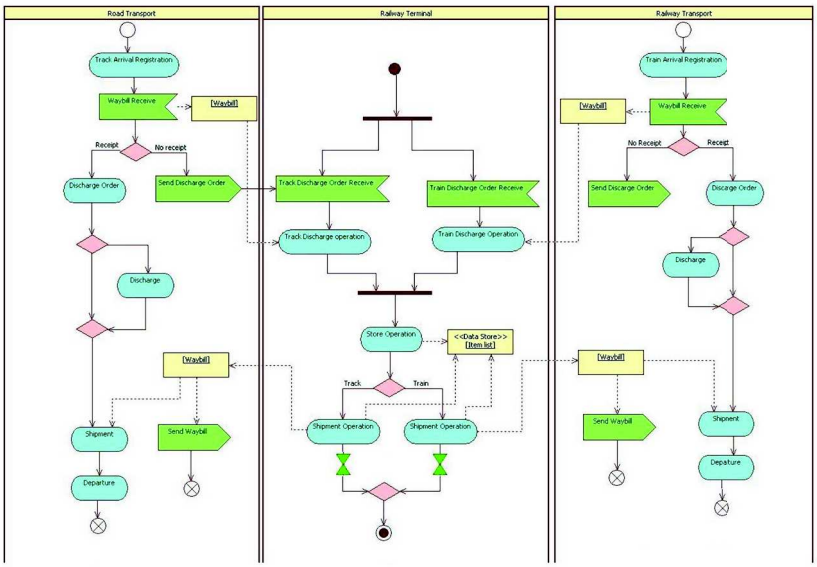
Each of three means of transport is described by business process in BPM Notation. Figure 6 presents exemplary model of the Train Terminal node modelled in BPMN. Road Transport Part, Train Terminal and Train Transport Part are identified as pools – lanes. A single connection between two terminals can be described by one or more processes, depending on number of phases of transportation stage.

Figure 6. Business process model of the terminal node using BPM Notation



The Links nodes (events) make possible to connect models of another terminals. This model focuses on the flow of business processes and relationships between them. Information processes such as the receiving and sending waybills and manage of storage containers is only signalled. Processes in the terminal node was also modelled using UML activity diagram (Figure 7). In this case the diagram is focused on processes of registration and processing.

Figure 7. Business process model of the terminal node using UML activity diagram



2.2. Example of modelling of the inbound logistic system

The architecture of inbound logistics system is organized into smaller parts as sub-systems and analytically developed. Analysis and identification of BPM requirements for logistics system is the first step in the modelling. The following example concerns the production logistics. The production logistics concerns logistic processes within a value adding system. It aims to ensure that each machine and workstation receives the required product in the required quantity and quality at the required time. The essence of the concern is to streamline and control the flow through value-adding processes and to eliminate superfluous processes. Production logistics can operate in existing as well as new manufacture.

BPMN diagram as a production model at its highest domain level represents fulfilment of the work order as responding to the received customer

order. Sales and Quality Departments are identified as separately black box lanes, Manufacturer as abstract pool with Production Department and Shop floor as lanes. In the Manufacturer pool the sequential flow of processes is mapped out through connecting them from a start event up to an end event using happy flow technique (Figure 8). The production process is triggered by Work order "Start" event message from Sales department. Collapsed sub-process of PP requirement in which a "plus" sign in the lower-centre of the shape indicates that the activity is a sub-process and has a lower level of detail (Figure 9a). Respectively the formation of sub-processes also applies to the activity as a process: Production Planning, Set-up Operating Process and Operating Process (Figure 9a, b, and c).

The PP requirement process followed by Production Planning Process leads to the Updating Production Plan System task. The production plan and the list of requirements is sent to Shop floor, Quality Department and Warehouse by Issue production plan picking task. This task is terminated with a "Message" end event. Figure 9 illustrates mentioned, collapsed sub-processes. In the next step, the model in first happy flow is supplemented by alternative paths, exceptions, activities and required artefacts.

Figure 8. The BPMN diagram model of Production System

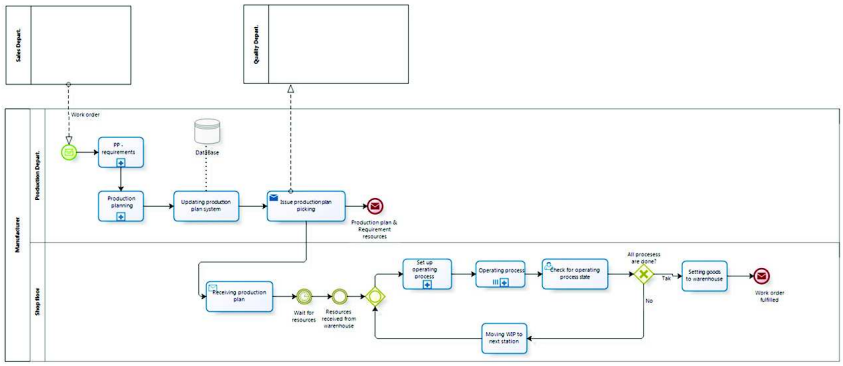
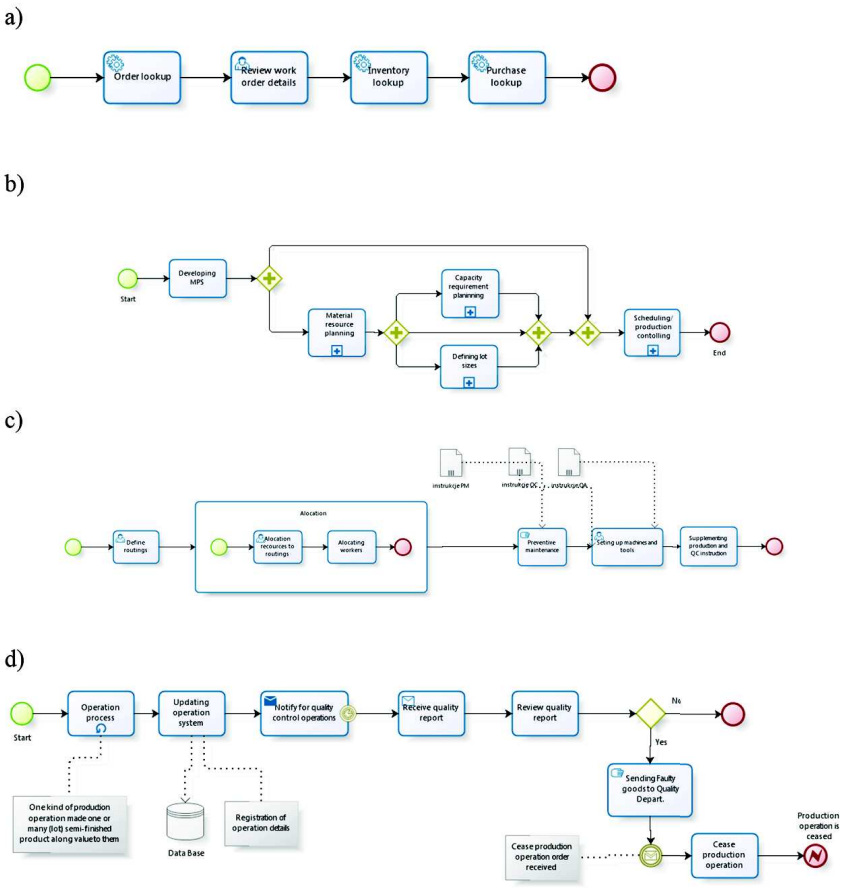
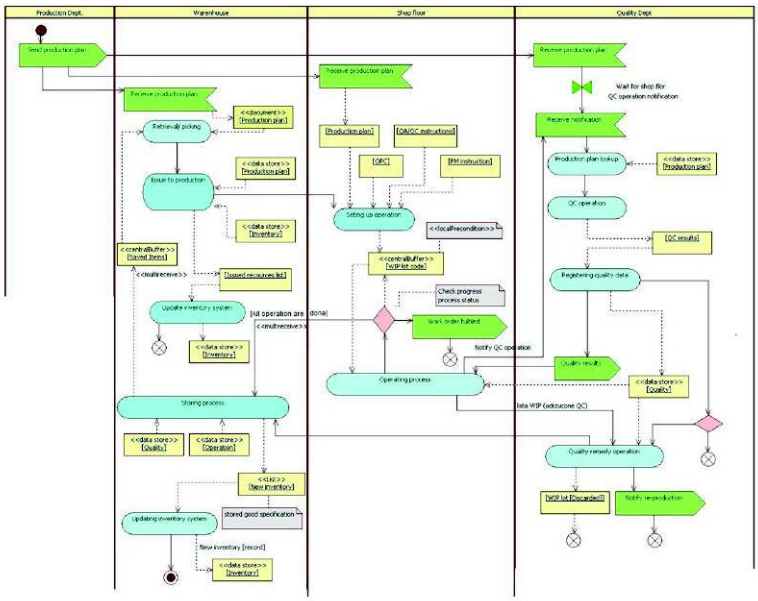


Figure 9. Expanded sub-processes: a) PP requirements; b) Production Planning; c) Set up Operating process; d) Operating Process



The UML activity diagram make possible modelling of the complete scenario of material transmission and transformation up to the finished goods. It focuses on information and physical object flow. Data on current production are collected in the facility Saved Items <<centralBuffer>>. Details of the Production Plan and Inventory are stored in objects <<datastory>>. The UML activity diagram presents coded data used directly to control production and the OPC data used as a standard of data exchange in the industrial automation space and in other industries.

Figure 10. The UML activity diagram as a meta-model of WIP (Work in process)



Source: Own study Khabbazi et al. 2013a.

Conclusions

The article describes the use of BPMN diagrams and UML Activity Diagrams for modelling logistics processes. UML Activity Diagrams are focused on the presentation of information flow and resources taking into account the object-oriented features of the process. In contrast, BPMN diagrams focus on the processes taking into account the events determining the runs and branching of these flows. The main objective of UML is modelling the system during its design.

Logistics processes are parallel movement of resources, whether in external or internal (eg, manufacturing) area and providing information about the movement, as well as ensuring the proper management of this movement. BPMN although not distinguish between the type of transmitted messages, it separates the content of a business process from the same information.

Coexistence of information processing with the course of business processes is the essence of modern management systems. Sometimes, however, often causes confusion in the design and modelling. The appearance of the BPMN diagrams is their significant advantage. In contrast, the UML activity diagrams are created separately for the model of the environment and to model the system.

UML activity diagram focuses only on the part of the activity of the business processes that involve the processing of information, such as the preparation of an encoded list of activities and elements of production. There introduces on the activity diagrams also detailed information about the input data and output data of each activities.

The utility of both types of diagrams confirmed by numerous publications in the field of business modelling and a variety of tools to support the creation of such diagrams. For example, a commercial system Enterprise Architect enables you to create both diagrams in UML and BPMN. There is also a free tool or tools in the free version. In many such tools for example should be mentioned: Star UML, ArgoUML, BizAgi Process Modeller. In addition to the possibility of a graphical representation of diagram elements, these tools have the most options to verify their correctness and for BPMN diagramming tools can simulate the processes. In this second case, the BPMN diagram is translated into a language of business modelling: BPML or SRML (Simulation Reference Mark-up Language). It is also possible to transform BPMN diagram to BPEL4WS script to create network service system (Badura 2012).

References

1. Aalst van der, W.M.P. (1999), Formalization and verification of event-driven process chains, Information & Software Technology, Vol. 41, No. 10, pp. 639-650.
2. Aalst van der, W.M.P., Hofstede ter, A.H.M. (2002), YAWL: Yet another workflow language, QUT Technical report, FIT- TR-2002-06, Queensland University of Technology, Brisbane.
3. Aalst van der, W.M.P., Hofstede ter, A.H.M., Kiepuszewski, B, Barros, A.P. (2003), Workflow patterns, Distributed and Parallel Databases, Vol. 14, No. 1, pp. 5-51.
4. Badura, D. (2012), Modelowanie aplikacji i usług internetowych, in: Pikiewicz, P. (Ed.), Zastosowania Internetu, Wydawnictwo Wyższej Szkoły Biznesu w Dąbrowie Górniczej, Dąbrowa Górnicza.
5. BPML.org. Business Process Modelling Language (BPML), available online at: www.bpmi.org (accessed 1 November 2002).
6. Khabbazi, M.R., Hasan, M.K., Sulaiman, R., Shapi'I, A. (2013a), Business process modelling in production logistics: complementary use of BPMN and UML, Middle-East Journal of Scientific Research, Vol. 15, No. 4, pp. 516–529.
7. Khabbazi, M.R., Hasan, M.K., Sulaiman, R., Shapi'I, A. (2013b), Business process modeling for domain inbound logistics system: analytic perspective with BPMN 2.0, Journal of Basic and Applied Scientific Research, Vol. 3, No. 9, pp. 569-578.
8. Koniewski, R., Dzielinski, A., Amborski, K. (2006), Use of petri nets and business processes management notation in modelling and simulation of multimod-

al logistics chains, Proceedings from 20th European Conference on Modelling and Simulation – ECMS.

9. OASIS (2007), Web services business process floor execution language version 2.0. committee specification.
10. OMG (2007), Unified modelling language: superstructure, Technical report, version 2.1.1.
11. OMG (2011), Business process model and notation (BPMN 2.0), formal/2011-01-03, available online at: http://www.omg.org/spec/BPMN/2.0.
12. Petri, C. (1962), Fundamentals of a theory of asynchronous information flow, IFIP Congress, pp. 386-390.