The importance of an ontological view of a business system for the correct design of its business processes

Václav Řepa[1,2]

[1] Prague University of Economics and Business, Faculty of Informatics and Statistics, W.Churchill sqr. 4, Prague 3, Czech Republic.
[2] Škoda Auto University, Na Karmeli 1457, Mladá Boleslav, Czech Republic.

Abstract

In the paper we explain the idea that an essential condition for the correct design of the system of business processes is the careful respect of the ontology of the business system. Therefore, the basic source for the design of the business process system is the conceptual analysis of the business system to which the business processes belong. Such an analysis provides important information about general business aspects that must be considered in all business processes. We briefly introduce the MMABP methodology, which is based on the presented idea, and show the basic methodological consequences of a necessary respect for the ontological substance of the business system in the process system design. We also show and discuss typical mistakes and problems of business process design related to a low respect of the business system ontology and illustrate some of them with a practical example. Finally, we summarize the main ideas of the paper and outline some important challenges for the future development.

Keywords

conceptual modeling, ontology modeling, business process modeling, MMABP.

1. Introduction

Proper design of a process system is a critical point in building a business system. Business process design is not only a technical issue, it must lead to the application of the ideas of a process-driven business system.

The basic idea of a process-driven business system is excellently explained in [1]. The authors argue that it is necessary to make the enterprise flexible enough to be able to immediately exploit the opportunities permanently created by the technological development. According to the authors, the main value of technological development is that it allows to "do things differently", i.e. to change business processes. Therefore, it is necessary to base the management of the company primarily on its business processes. Business processes must play the role of the essence of the company's management, which requires being able to change the company's behavior (i.e. business processes) as soon as the technological opportunity appears. To do this, the company must maintain the definition of its processes as a set of models that govern the work of all infrastructures and make the whole company (including all its infrastructures, both technical and organizational) flexible to the opportunities of change. In this way, the company has established a "living link" with the development of technology and has fulfilled the idea of automating as much as possible.

Although the concept of a process-driven organization is widely accepted, its fundamental idea of a flexible enterprise is often disregarded. For example, let us consider the concept of "full automation" of business processes, which is often mentioned by technically oriented authors. The concept of a flexible enterprise shows that the idea of "fully automating business processes" contradicts the idea of process-driven management. Once "fully automated", the process is no longer intended to be a subject of change and becomes just a single step, a black box in the model. Such piece of the company's behavior is no longer relevant in a process-driven management context. Process-driven management requires a clear definition of the business process system, in which each process

BIR-WS 2025: BIR 2025 Workshops and Doctoral Consortium, 24th International Conference on Perspectives in Business Informatics Research (BIR 2025), September 17-19, 2025, Riga, Latvia.
repa@vse.cz (V. Repa); vaclav.repa@savs.cz (V. Repa)
0000-0001-9113-3447 (V. Repa)

represents a set of activities that are essential to achieving specific business goals and therefore, must be a subject of potential improvement / change. This essentiality corresponds naturally to the ontological substance of the business system.

Since process-driven management is the way of continuous penetration of the company with technologies, the process system must meet strict technological requirements. For process models, this means, among other things, fulfilling all necessary attributes of the algorithm. This requirement has a significant impact on the organization of processes in the process system. Each process in the system must be a single algorithm. Any natural parallelism between business activities should therefore lead to their placement in different processes[1]. This fact significantly influences our decisions in designing the structure of the business system, whose many structural features are therefore generally necessary.

In addition to the necessary attributes of the algorithm, the process system designer should also respect general rules for the contexts between the process structure and an ontological structure of the business system. The idea of such context comes from the work of M.A.Jackson first published in [2] as the context of a program and its data, and then generalized to the system structures in [3]. Actually the same context, just seen from the different perspective, can be found also in the data structures normalization technique [4], [5], [6]. Both of aforementioned sources have been used in the MMABP methodology [7], on which the ideas in this paper are based. These sources are the foundation for the concepts of "structural consistency" and the "process normalization technique". In MMABP, this idea is generalized to the *context between the ontological structure of the business system* (represented by the conceptual model in terms of Guizzardi's OntoUML [8]) *and the structure of the business processes in that business system*. This context is the main subject of this paper.

The main idea of the paper is not about the mapping between some existing process and ontology concepts, but rather about respecting the ontology when structuring processes (i.e., the ontology dominates the structure of the process system). Consequently, by "design of business processes", we do not mean their implementation in terms of software design, but rather the creation of a process system model consisting of mutually interconnected business process models. The main research question that this paper aims to answer is "*How can the business he business system ontology determine the overall conception of the business process system, and why?*".

In the following section, we present the background methodology to provide context for the ideas presented. In the third section, we provide an overview of related work, which allows us to more precisely specify the main focus of the paper. Using the example in the fourth section, we demonstrate how the business system ontology, in the form of a conceptual model, can determine the overall conception of the business process system — a process map. In the fifth section, we discuss the relationships between the business system ontology and the process map presented in the example. We then generalize these relationships as reasons for structuring processes and support them with additional arguments in a separate subsection. The conclusions section then summarizes a generalized ideas, reflects them in the light of the research question, and outlines some future development challenges.

2. Background

The ideas presented in this paper are based on the principles of MMABP – Methodology for Modelling and Analysis of Business Processes [7]. MMABP is a methodology for modeling business systems. In its nearly 30 years of existence, MMABP has evolved from focusing solely on business processes to focusing on the complete organizational model, i.e., a business system. As a business

[1] It is good to mention here that most business process modeling languages, including the standard BPMN, allow specification of parallel activities (tasks) in the process description, which is methodically inappropriate. In BPMN, this is probably related to the fact that it does not take into account the need to model not only the internal algorithmic structure of the process, but also the global view of the process system, usually called a "process map". Specifying "parallel tasks" in the process structure is a perfect substance for various errors such as deadlocks, multiple-worlds assumption, and other meaningless and therefore useless process constructions. Only a small subset of possible process constructions that use internal process parallelism are safely correct.

system, we mean any system of people, technology, goals, and related activities that is created to achieve specific business goals (such as enterprise, association, office, public body, etc.). As a business process, we mean a general prescription of the workflow that is precise enough to be supported and managed with maximum technological use, as opposed to a computer program or the intuitive way people currently do their work as it is often understood.

According to this concept, a business system comprises a set of mutually collaborating business processes that work collectively to attain particular business goals. The achievement of these goals is generally influenced by overarching rules and regulations of the environment in which these processes operate. We refer to this collection of rules as business rules. Additionally, the MMABP methodology places significant emphasis on the information system as an integral component of the business system. It recognizes that the information system serves as a model of the overall business system.

Figure 1: Characteristic of a business system [7].

An essential root idea of MMABP is that the fundamental nature of business is achieving goals within a given environment (see Figure 1). Based on this definition, two phenomena that form the basis of the MMABP framework for business systems modeling are identified: *intentionality* and *causality*. The so-called MMABP Philosophical Framework for Business Systems Modeling, which defines the basic principles and components of MMABP, is explained in detail in [9].

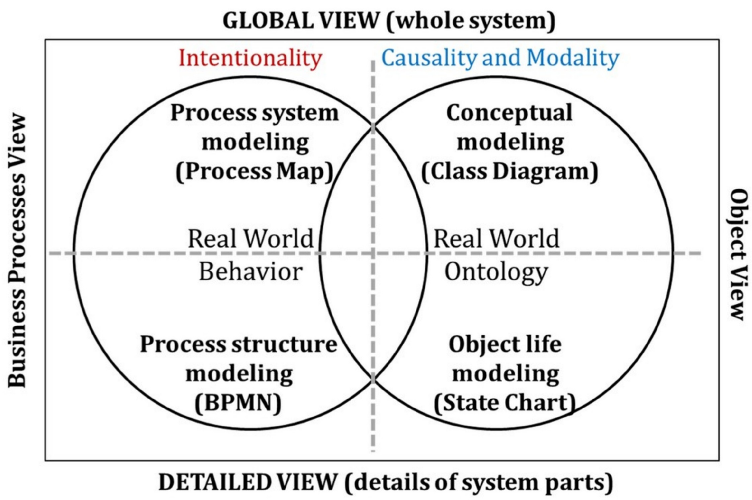

Figure 2: Illustration of the MMABP Philosophical Framework for Business System Modelling [9].

The MMABP philosophical framework, illustrated in Figure 2, defines four fundamental types of business systems models. These models encompass two fundamental views (system versus detailed) across two fundamental dimensions (intentionality versus causality/modality).

- *Causality/modality* is modeled by means of key two UML diagrams for the object description: The Class Diagram provides the global / system view (*conceptual model*), and the State Chart provides a detailed view of an ontological element of the system (*object life cycle model*).
- The *Intentional dimension* is modeled by business process models: A Process Map is used for a global / system view, and a BPMN model is used for a detailed view of a process element of the system (*process flow model*).

In addition to the two fundamental dimensions, MMABP acknowledges two fundamental perspectives.

- *Global alias System view*, in which the model describes the entire system as an organized set of elements (objects in the causality dimension, and processes in the intentionality dimension). The system view is always timeless; it cannot capture the temporal aspects, which can only be captured on the level of system elements.
- *Detailed view* describes the internal structure one system element. The detailed view always covers temporal aspects, such as business process flow in the intentional dimension, and object life cycle in the causal/modality dimension.

Business process models describe behavior within a business system, including goals, actions, communication, and collaboration. Object models, on the other hand, determine the business environment, including rules, conditions, constraints, and dependencies. These two fundamental dimensions must be harmonized within the business system, meaning business processes must unconditionally respect the system's rules. To achieve this harmony, which is sometimes referred to as equilibrium[2], MMABP establishes methodological principles and incorporates various consistency rules and techniques.

A particularly important feature of MMABP for the topic of this paper is that it regards the ontological and process views of the business system as tantamount, mutually complementing dimensions, rather than subordinating one to the other, as most current approaches do. Thus, according to the MMABP, the concept "**business system ontology**" refers in this paper to the ontology of the entire business system expressed in the form of a **conceptual model**. This model represents one of the two essential dimensions of the business system. From this perspective, the conceptual model represents the "**causal**" dimension of the business system. The second dimension is the "**behavioral**" dimension, represented by the **business process model**. Consequently, by "business system ontology," we **do not** mean the business process ontology in terms of the business process meta-model or the global model of business processes, also known as the "process map".

3. Related work

An essential importance of an ontology modeling in the field of enterprise modeling is convincingly expressed in [10]. The authors also demonstrate the close relationship between conceptual modeling and ontology modeling. This concept establishes an important connection between traditional, primarily theoretical ontology modeling and enterprise modeling. It has been elaborated on for the industry standard UML in Guizzardi’s OntoUML [8]. Another essential work that emphasizes the importance of ontology, primarily from the field of enterprise modeling, is Dietz’s well-founded DEMO methodology [11].

Since this paper primarily focuses on the design of business processes in a business system, the aforementioned resources, while generally important, are not specifically focused enough on business processes. There are a number of other resources that focus on an ontological view of business processes and are at least partially relevant to the this paper’s topic: [12], [13], [14], [15], [16], [17], [18]. Since there are several points of view on the relationship between business ontology and processes, none of these resources are entirely relevant to the idea presented in this paper. The

[2] We believe that the harmony of intentions and system modality/causality is not a one-way issue, i.e., harmonizing business processes with system rules. The rules of the business system express objective truths as well as specific internal rules that support business goals and intentions. Therefore, we speak of equilibrium rather than harmony.

focus of some resources is exclusively on the ontology of the business process model (e.g., [18]), a subject that is not pertinent to the theme of this paper. Such a focus actually leads to the business process meta-model, a general model of the process modeling domain. Some resources also consider a general ontology of the business system but always only partially, usually in the context of the process ontology. An approach based on a clear distinction between a business process system and a business system ontology, as in MMABP, could not be found in the available resources.

In [12], the authors discuss the mapping of business process models to OWL-S Ontologies. This approach is pretty close to the general business process meta-model, nevertheless specialized to the domain of web services. So, the paper uses particular language BPEL4WS and the particular OWL-S ontology and offers the mapping tool BPEL4WS2OWL-S.

[13] presents a general business process modeling ontology BPMO, which "is part of an approach to modeling business processes at the semantic level, integrating knowledge about the organizational context, workflow activities, and Semantic Web Services". In this approach, even if it is primarily focused on the business process meta-model, the authors have to take into account also some elements of the business system ontology. They specify concepts and their instances, some of them represent concepts from the business system ontology. Nevertheless, from the point of view of OntoUML conceptual model [8], they are able to capture only some concepts, and are not able to fully capture their relationships that exist there only related to the business process actions.

The paper [14] is also mainly focuses on a general ontology for business process modeling. The authors offer to integrate the Petri nets approach to the business process modeling with the ontology description language OWL and offer so-called "ontology based business process description". Including the ontology language naturally turns their attention to some business system concepts but the paper remains mainly focused on a general process modeling ontology (i.e. business process meta-model). Also in the paper [15], the authors introduce an ontological approach for modeling business processes. But the ontology is used in the paper clearly for the definition of the meta-model. Subject of modeling is the language itself, not a business system.

According to the claim: "Ontologies contribute to the conceptualization and organization of the embedded and unstructured information that is present in the business processes and that must be explored." it seems that the authors' approach in paper [16] might overcome the usual focus on a business process meta-model alone. However, instead of using the system ontology as the basis for possible process design, the authors propose mapping elements and knowledge extracted from a business process model in BPMN to an ontology. This approach contradicts MMABP's understanding of the role of business processes in the business system.

In [17] the authors introduce the ODD-BP approach, which combines the principles of semantic process definitions with a meta-model that implements a declarative and data-oriented process character. This approach is similar to the topic of this paper. However, the author's interpretation of ontology differs from that of MMABP, which views data as an implementation form of an ontology. They do not see such a relation between data and ontology, and take the latter in terms of a meta-model.

As it has been already mentioned, in both MMABP and this paper, we interpret ontology in terms of OntoUML as defined in [8]. The ontology is represented by a conceptual model that expresses the modal logic of a business system.

4. Example - Production business domain

Our example describes the processes in the company's Production business domain. The Production domain supports the primary function of the enterprise: producing various products according to customer orders.

Figure 3 shows a simplified fragment of the conceptual model that describes the ontology of the Production business domain. The company produces different types of *Products*. *Customers* can order a set of specified quantities of different products in one *Order*. Products are produced on the *Production Line*. The company has several production lines, and each line can produce several different products. Each product can be produced on at least one production line, some products on several lines. Production of each product requires cleaning of the production line and its preparation for the production of the given product, which significantly delays the production and increases its cost. Therefore, the company tries to optimize the production by keeping all requests for production of a particular product, derived from the existing customer orders, in a separate *Requests Queue* in order to achieve the largest possible *Production Batch* that can be produced all at once without the need to prepare the production line for another product. Optimization is based on a complex decision including the size of the production batch, mutual priorities of customer orders, number of running production lines, structure of individual orders and their production deadlines, and other important factors. The goal is to achieve the most efficient production with simultaneous satisfaction of customer orders in all parameters.

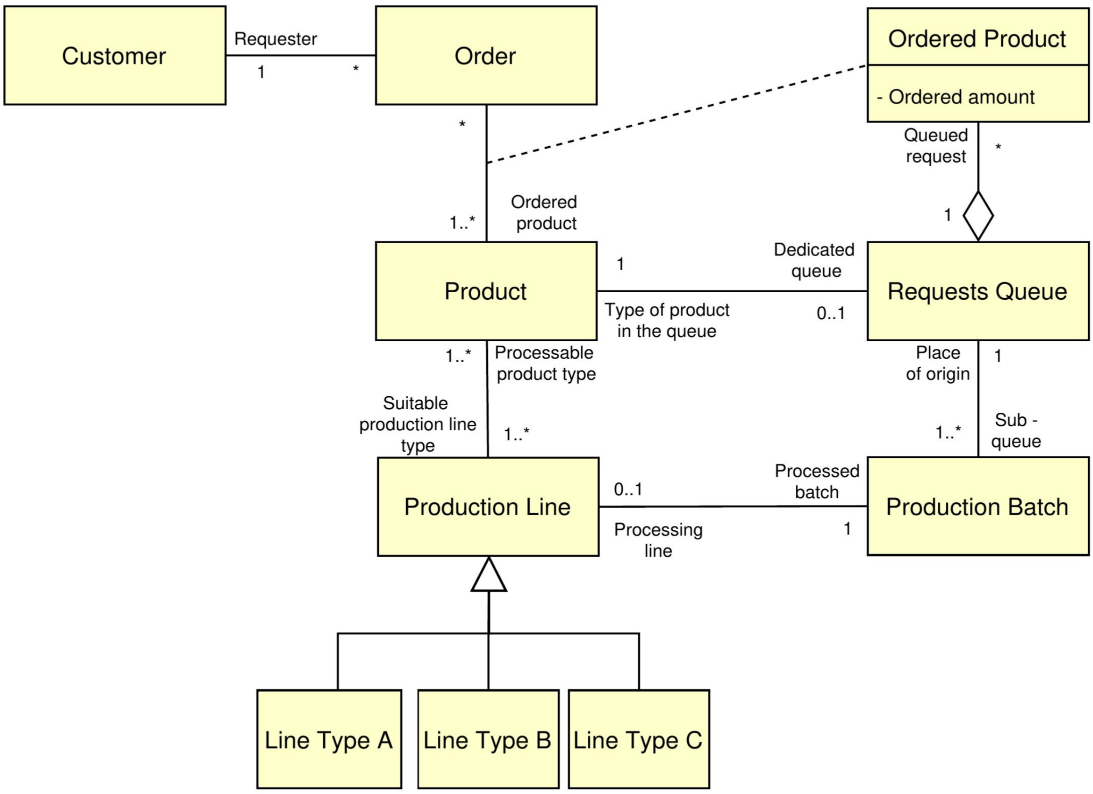

Figure 3: Fragment of the Production domain conceptual model.

Figure 4 shows the process map of the Production domain together with its internal customer - the *Customer Order Management* process, which is actually a key process of the company. This process addresses the entire business case for creating value for the customer, starting with the customer's request and hopefully ending with the customer's satisfaction. In one particular step, this process requires the service from the Production domain - production of ordered products.

The production domain consists of eight interrelated processes (see Figure 4). The local key process[3] in this domain is the *Production Request Management* process, which addresses the entire business case for creating value for the domain's customer - the *Customer Order Management* process. The *Production Request Management* process begins with the *Order Production Request* and ends with the *Production Result*, which represents all possible outcomes, both positive and negative. The process receives *Order production requests* from various instances of the *Customer Order Management* process, breaks each one down into specific products, and creates corresponding *Product production requests*

[3] A key process represents the value that a domain creates for its customers. Therefore, every domain has at least one key process. This process manages communication between the domain and its customers.

for the corresponding request queues. After that, it repeatedly waits for the responses from contacted instances of the *Requests Queue Management* process and collects the final result of the original order production request. It must also deal with any non-standard results, such as rejection of the request or delay in production, etc., and communicate them to the customer process (*Customer Order Management*). If the queue for the required product does not exist, this process has to create the new one (i.e. the new instance of the *Requests Queue Management* process). There is one instance of the *Production Request Management* process for each customer order.

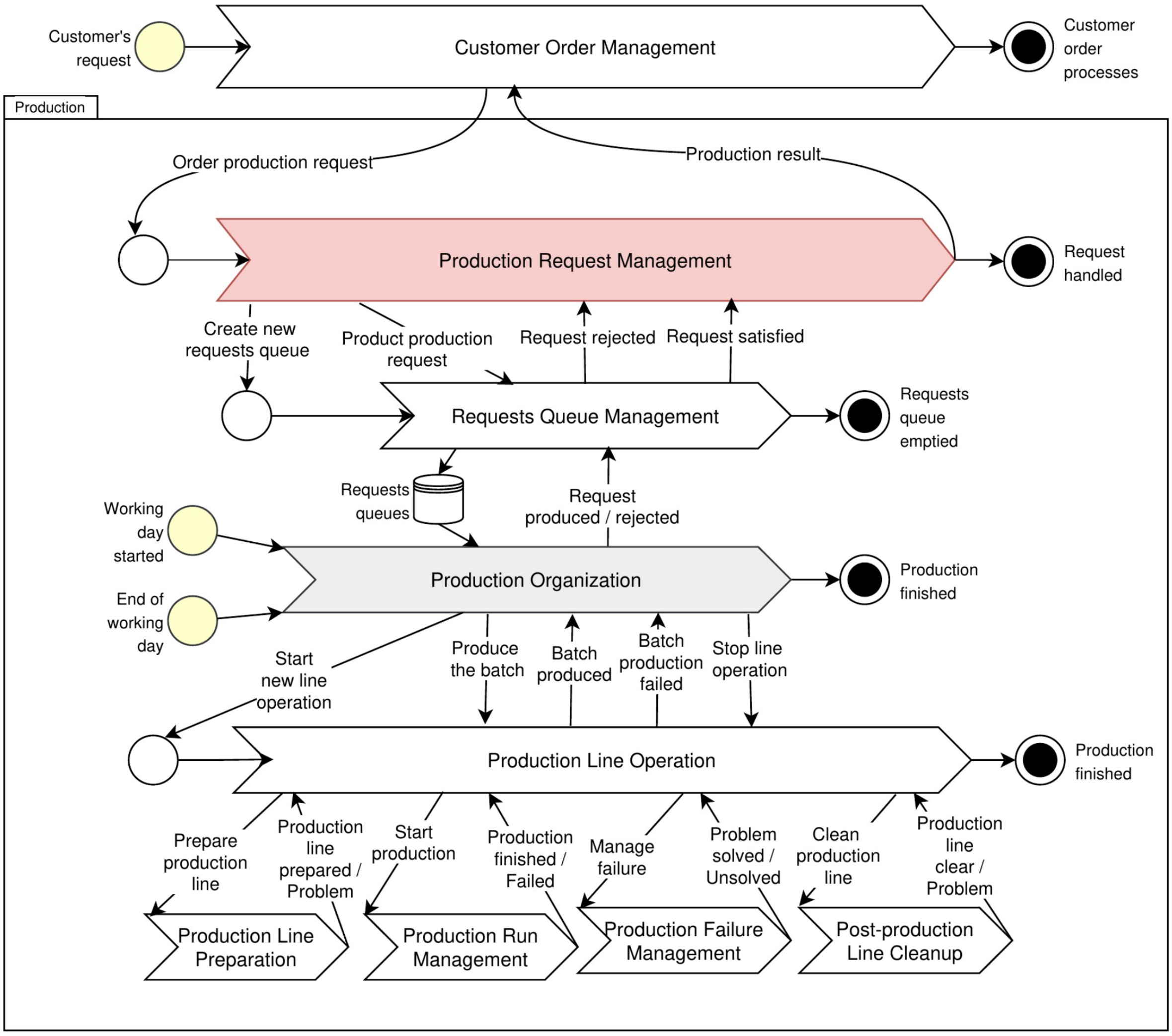

Figure 4: Process Map of Production business domain.

The *Requests Queue Management* process manages the queue of requests for production of one product. It receives the *Product production requests* from the *Production Request Management* process and sorts them in the queue according to their priority, urgency, and other parameters. It also receives the information about the result of the production of the request from the *Production Organization process*, and reports it to the *Production Request Management* process. There is one instance of the *Requests Queue Management* process for each queue of requests for the production of a product.

The *Production Organization* process is the main managing process of the Production business domain, responsible for optimizing production. The process periodically monitors the request queues, creates production batches from them, and sends the requests for their production to the production lines. If necessary, it starts the operation of a new production line or stops the operation of the line. It also receives the events about the result of the batch production requests from each production line: *Batch produced* or *Batch production failed*, decomposes the information into original production requests and reports the result (*Request produced* or *Request rejected* events) to the corresponding instance of the *Requests Queue Management* process. The *Production Organization* process periodically makes a crucial decision about the size and content of the production batch, taking into account the parameters of the production requests (i.e. the parameters of the original orders from which they come), the currently running production lines, their operation parameters (such as the time and cost needed to set up the line for each product), the possibilities of running a new line, etc. There is just one instance of the *Production Organization* process at a time. It starts at the beginning of the workday and ends at the end of the workday.

The *Production Line Operation* process manages the operation of a production line. The process is started by the *Production Organization* process and ends in response to the Stop line operation event generated by that process. The process uses the services of its local support processes, which address all three major phases of the production line operation: *Production Line Preparation*, *Production Run Management*, and *Post-production Line Cleanup*. It also uses the service of the supporting process *Production Failure Management* to solve a possible production failure. The process receives possible response events from these support processes: *Production line prepared*, *Problem with preparation of the line*, *Production finished*, *Production failed*, *Problem solved*, *Problem unsolved*, *Production line clear*, and *Production line cleanup problem*. There is one instance of the process for each production line in operation.

5. Discussion

Figure 5 shows how are business processes related to the business ontology. *Production Request Management* process is responsible to the *Customer Order Management* process for handling the production of the ordered goods. The process instance is related to the customer order. The conceptual model shows that several *Products* may be ordered in one *Order*. Thus, the process must also handle the one-to-many relationship between *Order* and *Product*. The process's task is to divide the order into its constituent products, send the *Product production request* to corresponding instances of the *Request Queue Management* process, receive all events that inform the process of the status of the requests, and the reassemble the order from them and report the result to the *Customer Order Management* process. This process cannot handle the request queues because they are related to one or more orders, and the orders are related to one or more queues. This relationship cannot be managed by a single algorithm. Therefore, each queue must be separately managed by an instance of the *Requests Queue Management* process. The *Requests Queue Management* process manages the production requests queue for a single product. It receives a production requests from the *Production Request Management* process and continuously sorts them according to various criteria, such as order production deadline, order and customer priority, etc. It also receives information about the result of each request production and reports it to the corresponding *Production Request Management* process. The instance of the process is directly related to the *Requests Queue* concept (see the conceptual model). Therefore, it cannot manage, as a single algorithm, at the same time the production batches, as they are related to the requests queue in a many-to-one manner.

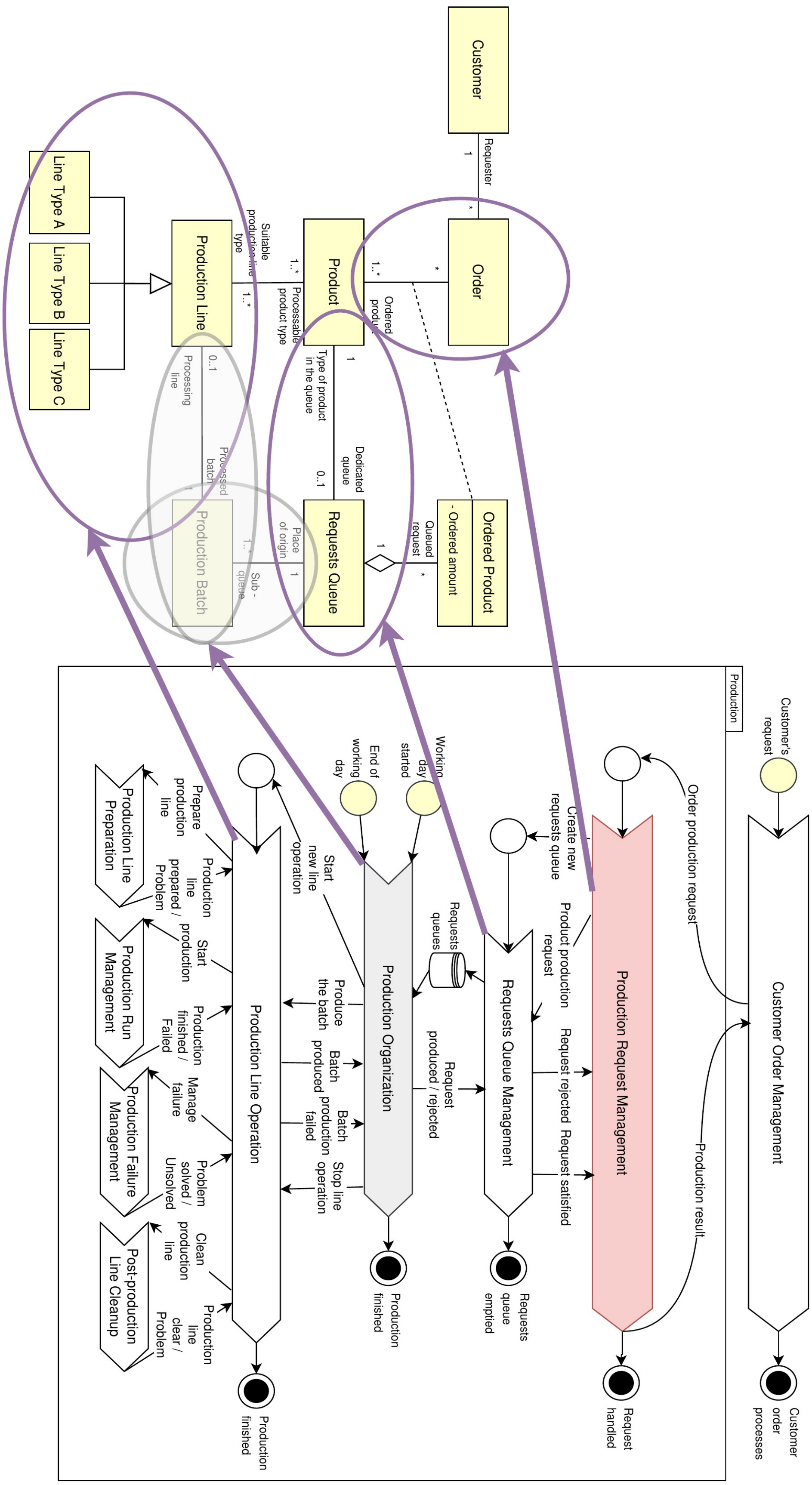

Figure 5: Relationships between conceptual model and processes.

The conceptual model shows that the *Production Batch* concept actually represents the relationship between the *Production Line* and *Requests Queue* concepts. This relationship is managed by the *Production Organization* process. The process continuously creates production batches from the request queues and places them on the production lines. This optimizes operation and satisfies production requests according to their attributes. Therefore, the process determines how many production lines are needed, which products each line will produce, and when each line must be rearranged to produce a different product. It is impossible to operate each production line in the same process. Therefore, the *Production Organization* process has to use the services of several instances of the *Production Line Operation* process. Each instance of the *Production Line Operation* process specializes on managing one production line. The process is related to the *Production Line* concept and all its possible sub-concepts, and also manages its relationship to the *Production Batch* concept in the conceptual model. It uses services of several supporting processes for individual typical parts of the line operation (see the process map), which makes it independent of their possible implementation or even possible outsourcing, thus contributing to the flexibility of the entire business system.

5.1. Reasons for structuring processes

As the example shows, there are serious reasons for dividing all the required actions into individual processes. The paper's main goal, derived from the research question presented in the introduction section, is to reveal the methods and origins of structuring processes within the process system in harmony with the business system ontology. We believe the essential reasons for structuring the processes are closely connected with the system ontology.

The reasons come from various complementary and partially overlapping sources.

Some of the reasons can be considered "**physical**" because they result from the physical parameters of the process system. For example, in our model, the Production Organization process cannot oversee the internal operations of the production line (e.g., preparation, operation, and cleanup) because there can only be one instance of the process that monitors all request queues in the system. This is because the process must make decisions about optimizing production for all products. Since each production line only produces one product, there must be multiple instances of the Production Line Operation process but only one instance of the Production Organization process. The physical difference between the concepts of "production" and "production line" necessitates distinguishing between these processes.

Another type of reasons for dividing actions into individual processes is "**evolutionary**". The process system designer must also consider its future natural evolution. The natural evolution of the process system directly follows the principles of process-driven management, which aims to make the company flexible enough to adapt to future changes, mainly driven by technological development. Any necessary changes should affect the least possible part of the system, possibly just one process. In that case, the change can be made in the easiest and safest way, without needing to change the rest of the system or, even worse, hard-wire the process logic into the hidden logical relationships of the different system variables' values.

The third type of reasons for dividing actions into individual processes we can call "**algorithmic**". MMABP requires, among other things, that a process model meet the definition of an algorithm. The basic definition of algorithm can be found in the standard [19]. In [20], Knuth defines five important features of an algorithm. These features are widely accepted:

- *Finiteness.* An algorithm must terminate after a finite number of steps. Infinite process would not be able to fulfill its final goal, which is a basic feature of a business process.
- *Definiteness.* In each step, the actions to be carried out must be rigorously and unambiguously specified for each case. This feature is a basic requirement for qualifying the process as technology-supported, which is another important aspect of process-driven management.
- *Having input(s).* MMABP requires an initial process input and at least one other input. These inputs, referred to as "intermediate" in BPMN terminology, represent process feedback in terms of cybernetics' "negative feedback," which was introduced in [21] as an essential

condition of purposeful behavior. For more information on the role of cybernetics in process modeling, see [7].

- *Having output(s)*. Outputs are physical representations of the process goal. Thus, the reasoning is the same as with the finiteness feature.
- *Effectiveness*. "Operations must all be sufficiently basic that they can in principle be done exactly and in a finite length of time by someone using pencil and paper". This request also reflects the necessary features of the business process related to the goal, such as finiteness and definiteness.

This strong reason is closely related to the evolutionary reasons for structuring the process system. In terms of the rules for algorithms, the designer automatically avoids possible parallelism in the process and is forced to perform necessary parallel actions in different processes. This situation occurs because these processes relate to different conceptual objects. Any future changes to the process system will always be related to a specific object. This rule thus ensures that activities are divided into processes in a way that supports localizing changes to a single process. We can illustrate how this algorithmic rule works at the example of *Production Request Management* process versus *Requests Queue Management* process in our process system in Figure 4. The *Production Request Management* process cannot manage multiple request queues because they are generated from various customer orders and require continuous sorting according to new incoming requests. Any attempt to express parsing a customer order into different request queues (a task of the *Production Request Management* process) and managing the ordering of requests in each queue with a single algorithm would necessarily lead to dividing this complex task into different algorithms. This is due to the need to perform mutually asynchronous operations simultaneously. This division also has significant evolutionary implications. Any possible changes to the organization of the queues are completely independent of the management of requests from customer orders.

To convince the reader about the importance and correctness of the idea presented in this paper, we argue inspired by Toulmin's model of argumentation [22]. For the illustration of the argument see Figure 6.

The **evolutionary reasons** for structuring business processes based on business system ontology, as presented in the previous section, highlight the importance of aligning processes within the system with fundamental business system concepts as straightforwardly as possible. This stems from the principles of **process-driven management**, particularly the need to structure business processes to maximize the system's flexibility. Flexibility of the process system means the ability to be easily changed. Therefore, any change to the system should affect as few parts of the system as possible. The most effective method for achieving this objective is to align the structure of processes with the inherent structure of business system concepts (i.e. its ontology) to the greatest extent possible. Of course, the structure of processes can never be exactly the same as that of system concepts, since processes should focus primarily on their relationships. Nevertheless, this can be done in many different ways, some more closely aligned with the conceptual system's structure than others. The farther the structure of processes is from the structure of system concepts, the more concepts can be secondary affected by even a simple change in the process.

The system's flexibility is also required by **technological development**, since the only way to take advantage of the opportunities it creates is to "do things differently" [[1]] (i.e. change business processes).

On the other hand, **physical reasons** express the need to distinguish basic business concepts clearly and respect their essential relationships for different purposes. This need also stems from the principles of **process-driven management** for the purpose to structure business processes to ensure system stability. Process-driven management is based on balancing flexibility and stability to avoid the extremes of chaos and immutability. Therefore, not only flexibility but also the stability of the entire process system has to be ensured. The best way to make a process system stable is to respect the business system ontology, which represents the real world's genuine substance. This stability comes directly from its physical origin. Another reason to structure business processes based on business system ontology is **technological development**, which requires the standardization of processes within the system. Standardization is a vital condition for technological development. Standards are necessary because they ensure a return on investment in technology development. Therefore, all support processes — i.e., processes whose flexibility does not directly impact the overall flexibility of the business system — should be standardized as much as possible.

At the same time, technological development is also the source of the **algorithmic reasons**. In order to be supported by technology, process models must meet the basic requirements that come from the definition of an algorithm. The work of M. A. Jackson [2] and [3] explains the natural difference between an algorithm and a system of algorithms. Respecting this difference is key to avoiding potential problems with parallelism in processes. This difference stems from the natural dynamics of basic business system concepts and their relationships. Therefore, MMABP [7] pays exceptional attention to object lifecycles.

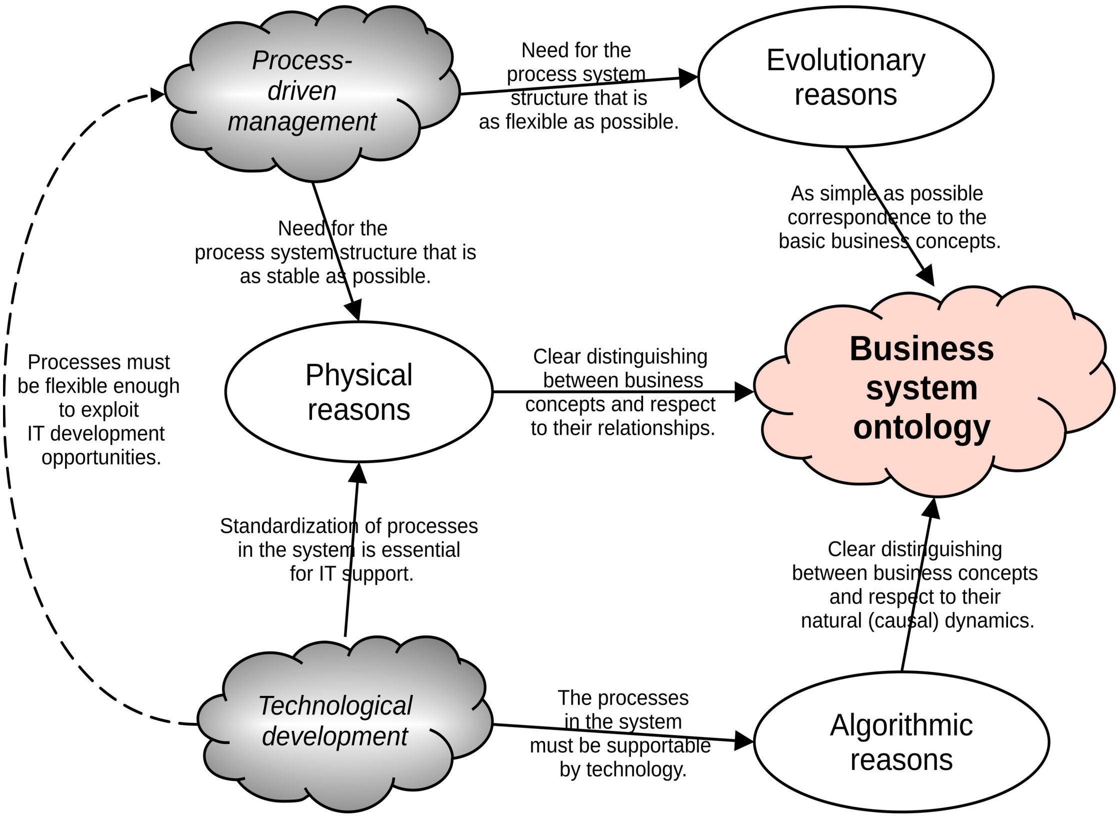

Figure 6: Reasons for structuring processes with respect to the business system ontology.

The above argument is based on two assumptions: the concept of process-driven management and the nature of technological development (see Figure 6). Without these conditions, the argument is not fully valid. For example, if someone does not consider the flexibility of the company to be part of the goal of process-driven management[4], then he or she can ignore the evolutionary reasons and the necessary relationship with technological development. This can significantly limit respect for the business system ontology. Ignoring the algorithmic features of the business processes as a necessary consequence of technological development[5] may lead to the process models that even contradict the business system ontology. These and similar exceptions, which are unfortunately not uncommon in practice, weaken the argument.

[4] This is, by the way, a typical feature of ideas about fully automated processes, RPA (robotic process automation), some approaches to Industry 4.0, and similar modern phenomena.
[5] As can be seen in some properties of the BPMN language.

6. Conclusions

The main research question, stated in the introduction section is "***How** can the business system ontology determine the overall conception of the business process system, and **why**?*". Therefore, in the previous section, we discuss not only the relationships between the business system ontology and its process system (related to the question of *how*) but also the reasons for respecting these relationships (related to the question *why*). From the discussion above, we can conclude that the reasons for dividing processes in the process system, whether physical, algorithmic, or evolutionary, all have one common denominator. They all stem from ontological distinctions within the business system. At the same time, the resulting structure of the process system adheres to an essential principle of process-driven businesses: the ability to adapt to future changes.

The most pressing challenges for the future development of the methodology are related to the application of neural networks (so-called artificial intelligence) in the real life. Our recent experiments on the use of large language models (LLMs) in business processes have repeatedly demonstrated the importance of correctly dividing activities into business processes using an ontology-driven approach, particularly for AI.

Acknowledgements

The paper was supported by the Faculty of Informatics and Statistics at Prague University of Economics and Business and by Škoda Auto University.

Declaration on Generative AI

The author has not employed any Generative AI tools.

References

[1] M. Hammer, J. Champy: Reengineering the corporation: a manifesto for business revolution. HarperBusiness, New York, NY, 1993.
[2] M.A. Jackson: Principles of program design. Acad. Pr, London, 1975.
[3] M.A. Jackson: System development. Prentice/Hall, Englewood Cliffs, N.J, 1983.
[4] E.F. Codd: A relational model of data for large shared data banks. Commun. ACM. 13 (1970) pp. 377–387. doi:10.1145/362384.362685.
[5] C.J. Date: An introduction to database systems. Pearson, Addison-Wesley, Boston, Mass. 2004.
[6] P.P.-S. Chen: The entity-relationship model—toward a unified view of data. ACM Trans. Database Syst. 1 (1976) pp. 9–36. doi:10.1145/320434.320440.
[7] V. Řepa, O. Svatoš: Fundamentals of business architecture modeling. Springer Nature Switzerland, Cham, 2024. doi:10.1007/978-3-031-59035-1.
[8] G. Guizzardi: Ontological foundations for structural conceptual models. Centre for Telematics and Information Technology, Telematica Instituut, Enschede, The Netherlands, 2005.
[9] V. Řepa,: Philosophical framework for business system modeling. In: 2023 IEEE 25th Conference on Business Informatics (CBI). pp. 1–6. IEEE, Prague, Czech Republic, 2023. doi:10.1109/CBI58679.2023.10187427.
[10] C. Welty, N. Guarino: Supporting ontological analysis of taxonomic relationships. Data & Knowledge Engineering 39 (2001) pp. 51–74. doi:10.1016/S0169-023X(01)00030-1.
[11] J.L.G. Dietz, H.B.F. Mulder: Enterprise ontology: a human-centric approach to understanding the essence of organisation. Springer International Publishing, Cham, 2024. doi:10.1007/978-3-031-53361-7.
[12] M.A. Aslam, S. Auer, J. Shen, M. Herrmann: Expressing business process models as OWL-S ontologies. In: Eder, J. and Dustdar, S. (eds.) Business Process Management Workshops. pp. 400–415. Springer Berlin Heidelberg, Berlin, Heidelberg, 2006. doi:10.1007/11837862_38.

[13] L. Cabral, B. Norton, J. Domingue: The business process modelling ontology. In: Proceedings of the 4th International Workshop on Semantic Business Process Management. 2009, pp. 9–16. ACM, Heraklion Greece. doi:10.1145/1944968.1944971.

[14] A. Koschmider, A. Oberweis: Ontology based business process description. In: Missikoff, M. and Nicola, A.D. (eds.) EMOI - INTEROP'05, Enterprise Modelling and Ontologies for Interoperability, Proceedings of the Open Interop Workshop on Enterprise Modelling and Ontologies for Interoperability, Co-located with CAiSE'05 Conference, Porto (Portugal), 13th-14th June 2005. CEUR-WS.org.

[15] T.-H.-H. Nguyen, N. Le-Thanh: An ontology-enabled approach for modelling business processes. In: Kozielski, S., Mrozek, D., Kasprowski, P., Małysiak-Mrozek, B., and Kostrzewa, D. (eds.) Beyond Databases, Architectures, and Structures. pp. 139–147. Springer International Publishing, Cham, 2014. doi:10.1007/978-3-319-06932-6_14.

[16] L. Riehl Figueiredo, H. Carvalho De Oliveira: Automatic generation of ontologies from business process models: In: Proceedings of the 20th International Conference on Enterprise Information Systems. pp. 81–91. SCITEPRESS - Science and Technology Publications, Funchal, Madeira, Portugal, 2018. doi:10.5220/0006709100810091.

[17] E. Rietzke, R. Bergmann, N. Kuhn: ODD-BP - an ontology- and data-driven business process model. In: Jäschke, R. and Weidlich, M. (eds.) Proceedings of the Conference on "Lernen, Wissen, Daten, Analysen", Berlin, Germany, September 30 - October 2, 2019. pp. 310–321. CEUR-WS.org.

[18] R. Singer: An ontological analysis of business process modeling and execution. (2019).

[19] ISO/IEC 2382:2015, 2015, https://www.iso.org/standard/63598.html.

[20] D.E. Knuth: The art of computer programming. Addison-Wesley, Reading, Mass, 1997.

[21] A. Rosenblueth, N. Wiener, J. Bigelow: Behavior, purpose and teleology. Philosophy of Science 10 (1943) pp. 18–24,. doi:10.1086/286788.

[22] S.E. Toulmin: The uses of argument. Cambridge University Press, 2003. doi:10.1017/cbo9780511840005.