Transdisciplinarity and the Future of Engineering
B.R. Moser et al. (Eds.)

doi:10.3233/ATDE220630

Object-Oriented Ontology Enterprise Architecture Framework Supporting Enterprise Knowledge Integration

M.H. CHAO[1], A.J.C. TRAPPEY and Neil K.T. CHEN
Department of Industrial Engineering and Engineering Management, National Tsing Hua University, Hsinchu, Taiwan

Abstract. In the era of Industry 4.0 and digital transformation (DT), the way the manufacturing industry provides value has shifted from selling products to directly creating value for customers. The synergy between business and information technology is the most important criterium for DT. This study proposed an object-oriented ontology-based enterprise architecture (OOOEA) framework, based on the unified modeling language (UML), to connect and integrate all databases and modules of the enterprise's existing information system (or ERP). The enterprise domain knowledge must be depicted to achieve business-IT alignment. This research starts from the physical and philosophical views of object-oriented ontology (OOO), pays attention to the essence of all things observed or understood in business operations, and constructs the core ideas of the OOOEA framework. The contribution comes from three aspects. One is to provide guidance for the first step of DT, the other is to propose a concise and applicable methodology as the basis for communication between employees at all levels, and the third is to verify the feasibility and effectiveness of the framework proposed in this research through a practical case study.

Keywords: Object-oriented ontology (OOO), knowledge engineering, business-IT alignment, digital transformation (DT), transdisciplinary engineering

Introduction

After entering Industry 4.0, the production method has evolved into a small number of diverse customized systems. The key to the competition between industry and service has also changed from the previous "price" to "customer value”. The way manufacturers provide value to customers is no longer as indirect and passive as in the past, but directly through digital technologies or system-based services. However, the success rate of digital transformation (DT) is not high. Many studies attribute it to the problems of "people" and "organizational culture". The issues at the management level are indeed complex. However, if an enterprise can find a simple and implementable method, starting from gradual improvement, finding a problem in the existing process and solving it effectively, and its quantified results can be demonstrated, it will be able to promote a positive cycle. So that companies will be more confident to ensure that they are on the

[1] Corresponding Author, Mail: s106034803@m106.nthu.edu.tw.

way to DT. The value creation mechanism of DT must be customer-oriented, domain-knowledge-oriented, and combined with digital technology capabilities.

The structural problem between the business operating model and information system is the main challenge faced by the transformation process and even the business operation process. Large companies generally accumulate structural problems between information systems and business operation models, which increases the difficulty of horizontal integration and shows the appearance of an information framework that lacks a blueprint for enterprise architecture (EA) [1]. Munir and Sheraz Anjum [2] sorted out many studies on the conversion between database and ontology, and pointed out that based on the complex interaction between the domain ontology and the entity relation model of the database, as well as the characteristics and limitations of the application, some works are necessary, including (a) define a semantic model of data, (b) specify domain knowledge, and (c) define links between different types of semantic knowledge. Despite the enormous potential of these data, it is mandatory to carefully plan and implement the whole process of data extraction, transformation and analysis to integrate the wide variety of heterogeneous sources, which cannot be done by only natural language processing (NLP) or the process of database and ontology transition [3].

To provide a not-so-complicated tool or methodology to demonstrate the EA, related approaches, including knowledge representation, ontology engineering, and databases, are investigated. For a DT to be successfully adopted, it is important that the methodology should be simple to use and do not require intensive learning. Three indispensable pillars making up the process of DT are evaluating, define the strategy, and implementing. First pillar, evaluation, are built by EA, the challenges of which come from two aspects: the importance and necessity of EA and valid tools or methodology. The purpose, goals and disciplines of EA are sound, but it seems the practice has not lived up to expectations. Simple tools with uncomplicated methodology, which are able to apply for completely depicting the EA and ontology of enterprise domain knowledge, should be established. DT is similar to business process reengineering (BPR) and must rely on the common understanding of the company's goals and participation of all employees. Employees at all levels of the enterprise have different areas of expertise and backgrounds. An enterprise-wide way of expressing EA is very important and challenging. After a comprehensive review of all the tools used to express the ontology, it is proposed that object-oriented ontology (OOO) is the most suitable point of view to explain the physics and reality, and it is also the most appropriate method to express the ontology.

1. Literature review

This research uses OOO as the core to promote the DT of enterprises. In the first paragraph, concept of OOO is discussed to support the novel viewpoints put forward by this research. Metaphysics focuses on the essence of realitas to explain all the things that can be observed or understood in the operation of the enterprise, thereby constructing the core ideas of this research and explain how DT activities can be promoted. DT is a change or can be said to be disruptive innovation, which is similar to the nature of BPR. Thus, relevant literature from BPR to DT are explored in the second paragraph so as to clearly understand the nature of DT. Since business-IT alignment is the main success factor of DT, relevant methods and frameworks are reviewed in the third paragraph. Furthermore, related works about knowledge engineering are reviewed in the last paragraph.

As the most important theoretical basis of this research, Harman [4] described in detail all the connotations of OOO, including concepts, basic principles, indirect relations, and approaches. This research intends to find a way that can fully explain all the operating rules and things of the enterprise from the aspect beyond human beings. The ‘theory whose range of applicability is limitless’ can only be found in OOO the type of philosophy [4]. From the perspective of quantified manufacturing in DT era, an automated new world is on the horizon: one comprised completely of entities to fully autonomous non-human automation. Speculative Realism is a contemporary philosophical movement that has forced itself to think about the world beyond its human correlation. Since Speculative Realism and OOO transcend the human gap, they are more apt to deal with issues that affect a posthuman world [5]. Specifically, in writing by philosophers and social theorists in the areas of speculative realism and OOO, the injunction to foreground the independent existence of entities is affirmed with particular force [6]. Other positive elements in OOO are its anti-scientism and anti-monism, which allow a richer perspective so that reality can be seen. The respect to objects allows them to be studied in genuine appearance while also considering their unknown sides or layers may show more possible essences [7]. The state of grasping objective facts is often called knowledge, and knowledge is taken to mean the human recognition of a truth, so that knowledge and truth generally come as a pair. OOO can be recognized as a discipline detecting the gap between knowledge and reality. Barely known to the public a decade ago, OOO has emerged in recent years as one of the most provocative philosophical theories influencing the arts, humanities, architecture [4]. IoT has been observed to have many applications for OOO [8]. As a researcher rather than a philosopher, this research and many of the above-mentioned studies are not discussing these philosophical speculations, but the core concept of the deconstruction of reality by OOO has a significant implication on the design pattern of computer science and data science.

Although the differences between BPR and DT are highlighted as that BPR is considered to focus on the rule-based process while DT on reimaging the process through IT and new data, some researchers and practitioners might see similarities between BPR and DT. In terms of definition, both aim to bring radical changes to the enterprise and regard them as one of the methods of enterprise transformation. In terms of implementation, BPR and DT both emphasize that creating value for end customers and the participation of all employees are the most important elements to succeed. Hence, it can be inferenced that DT is special case of BPR, which more emphasized on the use of digital technologies. This research starts from claiming 'organizational identity' and exploring the imposition and reconciliation in transformation activity accordingly, which indirectly promotes the idea of this research to explore the domain knowledge underlying DT in an "object-oriented" approach.

OOO aims at detecting the gap between knowledge and reality [4], just like this study aims at bridging the alignment gap between business and IT. EA offers a high-level overview of an enterprise’s business and IT systems and their interrelationships. Top four EA methodologies are the Zachman framework, the open group architectural framework (TOGAF), the federal enterprise architecture (FEA) framework, and the Gartner methodology. Studies have continuously integrated more frameworks, but the practicality of EA still cannot be widely proved. Although EA lacks a complete implementation, the concept of measuring business-IT alignment is still widely accepted [9], among which unified modeling language (UML) is widely used as a standard for describing the information system framework in EA [10]. UML’s wide recognition among software practitioners and its applicability in describing domain models make it possible for being a communication tool for employees at all levels of the enterprise [11]. As the easiest modeling language to learn and a basis for communication, although UML continues to evolve and adds more and more complex legends, only 20% of simple UML legends and notation are required for 80% of software usage [12]. Therefore, by selecting the most core element of UML, it should be able to provide an effective communication language for the enterprise.

On the way to DT, corporations of all classes are focused on the massive and optimal practice of information and communication technology in different organizational plans including human talent, organizational configurations, processes, inputs, outcomes, services and obviously the industry model. EA provides an industry 360-degree vision map and organization structure for business and technological changes. Knowledge engineering is used to represent knowledge and reasoning systems and organize the knowledge bases for fluid communication inside and outside the organization. Typical knowledge engineering tasks like acquiring knowledge and design captured in the heads of people and a further transformation into semi-formal representations that can be understood by people but that can also be processed (semi-)automatically by computer systems need high effort, which is a problem as it introduces cost, complexity, and other risks, and the conceptual model-based digital twins of design thinking artifacts must reflect the agility requirement [13]. An exploration of existential ontology in search of further clarification of the concept of thinging has some beneficial orientations in modeling. The incorporation of thinging in conceptual modeling is required to explain the roots of Heidegger's conception of things. This requires an understanding of Heidegger's existential ontology to identify any relationship to thinging [14]. Thus, this study falls within the intersection of two disciplines: software engineering / modeling / conceptual, and philosophy / ontology / existential. OOO merely rejects the idea of knowledge as a direct presence of reality itself, and does not scorn knowledge per se [4], even, because of this, it further proves the importance of knowledge engineering in OOO.

2. Object-oriented ontology-based enterprise architecture

The spirit and essence of OOO are the core concepts. OOO shows a firm philosophy of realism and materialism, which retains the concept of finitude between objects [4], which means that there is no single object can be fully understood by any other object, and each object interacts and has relationships with other objects from its own point of view, thereby staggering into rich business behaviors. In OOO's theoretical system, people are just one of the things in the world, and the rank of things should be raised to the same height, but this does not mean that the value of people is devalued. The rise of smart factories, IoT, and digital twins is definitely an important reason why the object-oriented nature must be taken seriously. In order to strengthen the mastery of all the machine equipment and workflow in the smart factory, the digital twin is established as the agent of the entity in the real environment in the virtual space. On the basis of the IoT, it starts from improving the visibility of information and further combining more data to integrated analysis and the development of remote control.

The spirit of object-oriented can be seen clearly in the embodiment of the smart factory: each machine or equipment can correspond to an entity object, and each entity object has its definition, properties, and behaviors. When all the individuals in a space can interact smoothly in the space from their respective angles, a large and complete system is formed. By OOO, the basis of the fanciful imagination of the future world can be a comprehensive object-oriented thinking and practice. DT emphasizes the ability of an enterprise or organization to quickly respond to changes. “The only constant in the world is change” is also a catchy slogan. Everyone understands this concept, but no one can come up with a specific way to put it into completely practice. This research proposes a methodology and complete framework for which OOO as the core concept to help enterprises face the changes and challenges in DT era.

The four layers and three aspects of proposed object-oriented ontology-based enterprise architecture (OOOEA) framework is shown in Table 1.

Table 1. Three aspects and four layers of object-oriented ontology-based enterprise architecture (OOOEA)

Aspects / Layers	Physic	Digital enterprise architecture (DEA)	Knowledge-based system (KBS)
Application	Workflows	Operations	Services
Business	Behaviors Activities	Business Processing Unit Data searching	Business intelligence (BI) Pivotable Data analysis
Domain	Entities Concepts	OOO	Data hub
Data access	Worksheets	Main database	Extensional database

In application layer, three oriented application scenarios are described from the perspective of enterprise users. For the physic aspect, the company conducts daily business operations through well-defined processes, which are composed of many activities. Corresponding to the DEA aspect, these activities may correspond to operations on information systems. For the aspect of KBS, employees at different levels of enterprises may have different data service request with related to their business content. Compared with the application layer attaches great importance to the user's interaction with the physical environment or system interface, the business layer describes the business domain insights of these interactions, including the operational behavior of the physic aspect, the business processing of the DEA aspect, and the data analysis of the KBS aspect. As drilling down to the domain layer, in the physic aspect, dynamic behaviors are decoupled into static objects and concepts, while OOO is constructed in the DEA aspect and the data is converted into meta data in the KBS aspect. From the perspective of knowledge engineering, the physic aspect corresponds to knowledge acquisition, the construction of the DEA aspect corresponds to the knowledge representation, and the data processing of the KBS aspect is for further knowledge utilization. The cross between business layer and domain layer and physic aspect and DEA aspect are the main scope of proposed Class-Activity-Status (CAS) model. Data access layer describes the database. In physic aspect, there may be some papery or electronic worksheets. In DEA aspect, classes in OOO are stored in various tables in the main databases. These tables record the raw data with the finest granularity. After the three steps of ETL (extract, transform, and load), the raw data from main database are converted and stored in extensional database to form a data warehouse. The data hub is the core of the domain layer in KBS aspect. It collects staging data from data warehouse and flash data dynamically assembled from OOO to form a large amount of meta data, which support diversified data analysis applications in business layer, including commonly used business intelligence and pivot table. Finally, through the provision of a service interface to meet the various needs of the application.

According to proposed OOOEA, several challenges concerned in applying domain-driven design to microservice architecture, including deducing microservices from domain models, missing infrastructure components in domain models, and autonomous domain modeling [15], are all included and can be explained as follows.

The first challenge, deducing microservices from domain models, uses the informality of the domain model to point out the impact caused by the unclear interface when providing microservices. Proposed OOOEA uses OOO as the main method of knowledge representation in the domain layer. Its purpose is to align with the actual user behavior of the physical aspect, and can be used as an object-oriented system model based on this, not directly as a service interface. This study argues that, since the informality of the domain model is to serve as the language of communication, it is inevitable that concrete interface operations cannot be unambiguously deduced from the model, which cannot be seen as a challenge, but a choice.

The following explains the second challenge: missing infrastructure components in domain models. The domain model must be the consensus of all employees of the enterprise. In the DEA aspect, the data access layer stores all classes in OOO in the main database with highly similar structure. From the KBS aspect, the main database data is transformed into the staging data stored in the extensional database through ETL. In addition, the flash data dynamically combined with the data in the domain layer's OOO are used to jointly create a data hub to support diversified data analysis requirements.

Finally, for autonomous domain modeling, service and domain model are both high cohesion and low coupling. The team responsible for service provision and the team maintaining the domain model may be independent, and may encounter challenges on concept integration. This challenge is practical. In proposed OOOEA, the purpose of distinguishing between business layer and domain layer is to provide solutions to this challenge. Through clearly distinguished layers, a clear boundary is provided for domain model access. On this boundary, changes are also restricted by clearly defined service interfaces. However, there is no restriction on the understanding of the domain model by all teams. It is precisely in order that the domain model can be widely understood and used as a basis for communication.

3. Class-Activity-Status (CAS) model

Figure 1 illustrates the OOO construction process, including three phases and two aspects. The three phases are knowledge acquisition, static structural modeling, and dynamic behavioral modeling. The two aspects are business process and OOO model. This study proposes a CAS model with three parts: class, activity, and status, which are intuitively associated with UML's class diagram, activity diagram, and state diagram respectively.

3.1. Phase 1: knowledge acquisition

At phase 1, starting with the description of the business workflow, it can be completed through the commonly used UML activity diagram. Activity diagrams are very similar to flowcharts, but are based on the creation and status changes of objects, while flowcharts are just drawn from the order of business operations. Two parts in this phase are essential: task categorization and input / output identification. For part 1, all tasks can be roughly categorized into three types: object construction, business operation, and analysis. For part 2, the input or output attached to all arrowheads must be clearly identified. After that, the flowchart must be evaluated for completeness and variability.

The construction process will enter phase 2 if the evaluation is passed. Otherwise, the flowchart for the business workflow description must be refined, and the task categorization and input / output identification must be reclarified.

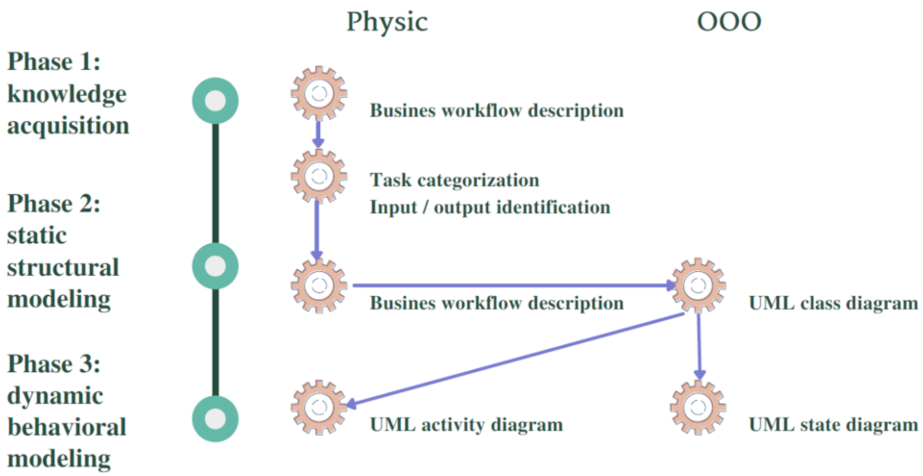

Figure 1. CAS model: object-oriented ontology construction workflow.

3.2. Phase 2: static structural modeling

Following the workflow obtained in phase 1, the input and output on the flowchart are divided into sheets, entities, or concepts, and after abstraction, the static structural model is depicted in UML class diagram. What follows is a recursive process, where the static structure of OOO is expected to gradually strengthen and become more stable by continuously confirming interpretability and reality. The UML class diagram components used in the static structure of CAS include class, enumeration, interface, and package. The relationship between components includes association, inheritance, realization, aggregation, and composition. According to the CAS specification, all relationships must be marked with multiplicity.

3.3. Phase 3: dynamic behavioral modeling

Phase 3 is about dynamic behavioral modeling, in which repeatedly coordinating activity validation and status design is the main task. OOO will not be formed until interpretability check is passed. Dynamic business processes and behaviors are represented by activity diagrams, and each activity in the process corresponds to the workflow level on the diagram. All activities in the process will correspond to the nature of business activities.

This study proposes the principle that: An essence will be realized by one or more activities, but multiple essences will not be included in an activity. The ambiguity of the essence of classes and the unclearly of business activities result in confusion of the status, so that the business model of the enterprise is constantly affected by the variability of the physical business activities. This phenomenon can be corroborated from many studies focusing on domain-driven design and microservice architecture in recent years. Each class in OOO may have multiple status definitions at the same time. In contrast, in flow and method design, there is usually only a single status for each class. From a behavioral point of view, linking all possible activities and trying to use a status definition to visit all paths, resulting in a complex status network and a complex status diagram, making it difficult to maintain and face the diverse changes. When looking at the same class from different viewpoints, different essential aspects might be seen, and different status definitions might be described. The purpose of OOO is to use easy-to-understand diagrams as a common language between business and information personnel. State diagrams are not new, but the multi-status concept is crucial for making the essential concepts of business domain knowledge be clearly expressed.

3.4. The relationship between the model and reality

Object-oriented modeling is already a very mature software programming method. CAS basically uses the tools used by OO modeling, that is, UML. The steps of the modeling process look very similar to OO design, but its connotation is essentially different: CAS is not modeling for software systems, so it does not aim to meet user s' requirement, but to construct the domain knowledge ontology which is in line with the reality of enterprise. The OOO constructed by CAS is for communication, so many details are discarded. In fact, although UML continues to develop more diagrams and more comprehensive notations, they will not be used in practical applications. With the transformation of the digital age, the requirements of users are constantly changing and even need to be created or developed, the dynamic capability to constantly and quickly respond to changes is required. The purpose of modeling is to communicate and draw the distance between developers and users, not to generate a large number of comprehensive documents. CAS can be regarded as a practice of the agile principle, but it has replaced the concept of demanders. In other words, the reality and its essential concept are the requirements to be satisfied by OOO, and the requirements of users or business departments are only the outward appearance of reality and essential. For this reason, this study proposes three stages of the CAS model: modeling the reality, verifying the model by requirements, and verifying the reality.

Stage 1: modeling the reality. Modeling reality is the first stage. Before doing this, all that the company has is a bunch of employees and the knowledge in their minds. This knowledge exists in an intangible way, but it cannot be used and communicated properly. These employees come from different departments, have different backgrounds, and at different levels within the organization, they have inconsistent views and language. In order to solve this problem, modeling reality is the purpose of this stage. It is not to clarify the existing standard operating regulations, nor is it to clarify the existing information system framework, but simply to clarify the understanding of the true status quo of all employees in the enterprise and model the reality.

Stage 2: verifying the model by requirements. The CAS model must be verified by all actual business behaviors. In the process of verification, the inadequacy of the model gradually appears and needs to be constantly adjusted. This is an iterative process until the stability of the model is revealed. To reach this stage, the essence of the model will inevitably be established. In the process of OO design and programming, it is a necessary condition to satisfy the requirements; in the CAS framework of OOO, it is a sufficient condition to satisfy the requirements. As long as the essence of OOO can be clarified, the requirements in all business behaviors will naturally be met directly or indirectly.

Stage 3: verifying the reality by the model. The last stage is to understand errors and conflicts in actual business behavior through OOO, thereby forming an opportunity for process improvement or BPR. Through the CAS model, it has taken the first step for the expression of domain knowledge in the operation of the enterprise, so that the domain knowledge that originally exist only in the minds of employees can be transformed into a clear and easy-to-understand format, promoting horizontal communication within the enterprise. When there are consistent tools and expressions that can present the current state of knowledge in all areas of the enterprise, it will be very helpful to observe the operation problems within the enterprise. When this stage is reached, the business process of the enterprise will be able to grow synchronously with the information service, forming a digital enterprise.

4. Conclusions

The necessity and complexity of DT have been fully recognized. However, there is still a lack of a complete framework description and specific and implementable methods for DT solutions. From the perspective of knowledge engineering and object-oriented modeling and ontology, this research proposes the first step that can be carried out for DT. Proposed OOOEA framework with core CAS model is a simple and feasible methodology from static structure, dynamic behavior, and status model, which focuses on the acquisition of knowledge, and proposes a set of knowledge expression methods that are easy to understand and promote, so that knowledge can be easily transferred and used within the enterprise.

The novelty of this study comes from two aspects:

(1) Enterprise knowledge representation: This research explains the issue of enterprise knowledge integration in DT from the core ideas and concepts of OOO, and defines the workflow of enterprise knowledge modeling through basic UML notation as a communication language for enterprises.

(2) Integration of enterprise knowledge and information system: Combining the widely used characteristics of UML in requirement engineering and software engineering, with the four-layer structure of enterprise architecture, the proposed OOOEA framework is used as a novel discipline to explain "the everything of enterprise".

For managerial implication on DT agenda, by OOOEA framework, the essential link between OOO from philosophy and OO programming from computer science may be established, and the pace of DT can be accelerated with the practice for building enterprise domain knowledge with CAS model. The transdisciplinary integration with philosophy, computer science, knowledge engineering, and management shows great importance and necessity.

The purpose of this research is to provide enterprises with guidance on DT, but the reality and architecture of enterprises vary, so there is no single criterion for evaluating the success of DT or the degree of business-IT alignment. The methodology of this study must rely on more case studies to verify its applicability and completeness, which becomes the limitation of this study and meantime points the direction for future research.

Acknowledgment

This research is partially supported by the Ministry of Science and Technology (Taiwan) individual research grants (Grant numbers: MOST-110-2221-E-007-113-MY3 and MOST-108-2221-E-007-075-MY3).

References

[1] R. Chang and K. Chen. *Enterprise architecture, the strategic thinking you need in the digital age*. Harvard Business Review Instructor Lecture 2020 2020/10/29 , https://www.hbrtaiwan.com/article_content_AR0010035.html.

[2] K. Munir and M. Sheraz Anjum, The use of ontologies for effective knowledge modelling and information retrieval. *Applied Computing and Informatics*, 2018, 14(2), pp. 116-126.

[3] D. Blazquez and J. Domenech, Big Data sources and methods for social and economic analyses. *Technological Forecasting and Social Change*, 2018, Vol. 130, pp. 99-113.

[4] G. Harman, *Object-oriented ontology: A new theory of everything*. Penguin UK, London, 2018.

[5] S. Recht and N. Wiberg, QUANTIFIED MANUFACTURING, LITTLE DATA AND THE NEW ÆSTHETIC. *CSPA Quarterly*, 2018(21), pp. 8-19.

[6] D.P. McCormack, *Atmospheric things*. Duke University Press, Durham, 2018.

[7] F. Ferro, Object-Oriented Ontology's View of Relations: a Phenomenological Critique. *Open Philosophy*, 2019, Vol. 2(1), pp. 566-581.

[8] J. Lindley, H.A. Akmal, and P. Coulton, Design Research and Object-Oriented Ontology. *Open Philosophy*, 2020, Vol. 3(1), pp. 11-41.

[9] D. Venkatesan and S. Sridhar, A rationale for the choice of enterprise architecture method and software technology in a software driven enterprise. *International Journal of Business Information Systems*, 2019. 32(3), pp. 272-311.

[10] M. Ozkaya, and F. Erata, A survey on the practical use of UML for different software architecture viewpoints. *Information and Software Technology*, 2020, Vol. 121, 106275.

[11] R. Wohlrab, J. Horkoff, R. Kasauli, et al. *Modeling and Analysis of Boundary Objects and Methodological Islands in Large-Scale Systems Development*. In: G. Dobbie, et al. (eds.) *Conceptual Modeling. ER 2020. Lecture Notes in Computer Science*, vol. 12400. Springer, Cham. https://doi.org/10.1007/978-3-030-62522-1_42.

[12] K. Siau, and P.-P. Loo, Identifying difficulties in learning UML. *Information Systems Management*, 2006, Vol. 23(3), pp. 43-51.

[13] M. Walch, and D. Karagiannis, Design Thinking and Knowledge Engineering: A Machine Learning Case. *International Journal of Machine Learning and Computing*, 2020, Vol. 10(6), DOI: 10.18178/ijmlc.2020.10.6.1003.

[14] S. Al-Fedaghi, *Existential ontology and thinging modeling in software engineering*, 2019, http://dx.doi.org/10.2139/ssrn.3360382.

[15] F. Rademacher, J. Sorgalla, and S. Sachweh, Challenges of domain-driven microservice design: a model-driven perspective. *IEEE Software*, 2018, Vol. 35(3), pp. 36-43.