Methods for Developing an Information Model for a Machine-Building Enterprise on the Basis of an Integrated Approach to Information Management*

Elena V. Shendaleva (0000-0002-9993-7105)[1(*)], and Olga A. Shendaleva (0000-0002-1940-9805)[2]

[1] Omsk State Technical University, Omsk, Russia
shendalevaev@yandex.ru

[2] Omsk State Transport University, Omsk, Russia
oa_shendaleva@mail.ru

Abstract. Software and hardware differentiation of information flows by areas of activities of a machine-building enterprise, as well as the temporary inconsistency of these flows, is one of the serious problems in managing such an enterprise. In this case, operational management is not linked to the strategic objectives of an enterprise, which generally affects its sustainable development. The construction and use of the enterprise information model serve as a tool for coordinating information flows. The study addresses topical issues of building an enterprise model based on the integrated approach to information management. Technologies formed a unified information environment are based on structuring and describing (modeling) various aspects of enterprise activities and creating its information architecture. Existing corporate information and management systems often have a limited scope that does not cover all enterprise aspects. This study aims to analyze the existing technologies for modeling machine-building enterprise activities and develop proposals for forming its information model. The authors analyzed the methodologies for creating an enterprise model as a combination of its technological, auxiliary, and business processes models. More than that, the authors compared software components written in the unified modeling language and object-process language. According to the results obtained, the authors proposed using the object-process methodology as the basis for creating an integrated or combined information model of a machine-building enterprise.

Keywords: Enterprise model · Enterprise information architecture · Object-process methodology

1 Introduction

The study focuses on constructing an opportunity analysis of an engineering enterprise model based on existing software complexes. This type of activity began to develop in the 1980s. However, due to the complexity of elaboration and insufficient development of information technologies, practical implementation of such models has narrowed to certain areas (e. g., business process management or service management). There is a renewed interest in the development of holistic enterprise models and the use of such models in enterprise management using artificial intelligence.

Theoretical approaches in this direction have been well studied and quite developed. Practical implementation, especially in Russia, faces difficulties at the initial stage, namely, choosing a direction and a developed model. The proposed study will enhance knowledge about the purpose and application of enterprise models.

Enterprise modeling is an abstract representation, description, and definition of the structure, processes, information, and resources of an organization. An enterprise is a complex system, which description consists of a model variety that describes the main characteristics of real objects, processes, and phenomena. These models and their components can be presented in the form of electronic documents, databases, and software files. An enterprise model is a representation of what an enterprise intends to do, how it works, and, possibly, how it is organized (Dori, 2002). Fig. 1 shows the structural diagram of the main activities of a machine-building enterprise. An information process that brings together all other activities is its central process.

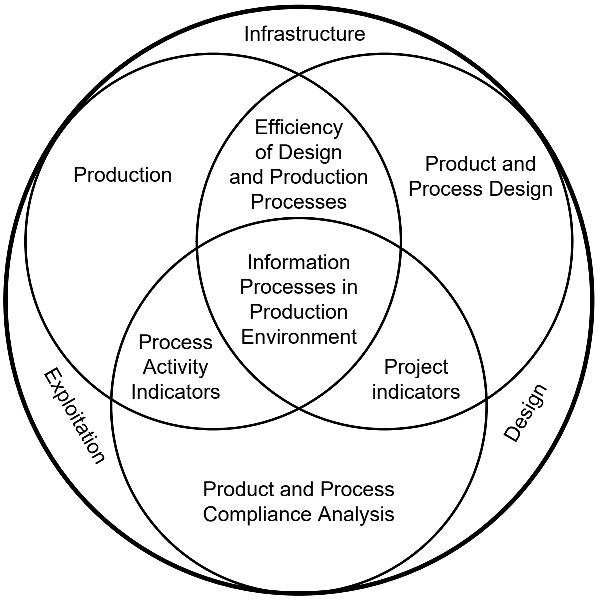

Fig. 1. Processes in the production environment. *Source*: Compiled by the authors.

A model of any kind can be reduced to an information model, that is, a comprehensive standardized representation of objects, processes, properties, parameters, and links in the form of information data sets. An enterprise information model can contain mathematical and process models, storage, management, and data exchange models, model transformation rules, and model graphical representations.

An enterprise model can function only in the integrated environment with certain information architecture. The integrated approach used to create an enterprise model is based on the following principles:

- All information in a model is presented according to the rules of the semantic network;
- Simulations should accompany an enterprise life cycle;
- Quality control of the decisions is performed using simulation models.

Enterprise models are used as tools for the systematic description of enterprise activities (Al-Fedaghi & Alahmad, 2017). They do not duplicate the enterprise life cycle, but they are limited approximations to the existing reality. Reference and simulation models are used as target models of an enterprise. A reference model is a structured set of interrelated information models covering an enterprise from a specific perspective.

Simulation models allow one to do the following:

- Form strategic development prospects;
- Perform dynamic analysis of possible development scenarios;
- Determine the impact of external and internal enterprise factors;
- Determine the efficiency and effectiveness of enterprise processes.

Enterprise modeling includes the following aspects (Mordecai & Dori, 2017):

- Methodological aspect defining formalization and structuring concepts of simulated systems, methodological foundations of system modeling, and approaches to the creation of stratified descriptions of simulated systems;
- Mathematical aspect related to the use of various statistical methods, optimization and decision-making methods, and artificial intelligence methods;
- Technological aspect defining the requirements to the software and hardware complexes that serve as a basis for the integrated approach in developing an enterprise model.

The integrated environment combines information from multiple independent sources into a single logically sequential data set. Implementing the integrated environment and creating an enterprise information model are possible based on the integration, unification, and consolidation of these model components.

Integrated models are characterized by using a standard model form based on a common template for presenting information, for example, according to the Information Resource Dictionary System (IRDS).

A general structure template of the meta-model is used for unified models since it allows transforming private models and establishing the semantic equivalence of composite models. Unified models are used, for example, to model enterprise business processes.

A combined model is created in a situation when it is impossible to establish the semantic equivalence of all private models. A certain degree of unification of private models and their dynamic and terminological harmonization are used in this case. A combined model is most likely for engineering enterprises in their current state when most private models are not standardized or in the unified form (Industrial automation systems – Concepts and rules for enterprise, 1998).

The basis of an enterprise information model is a conceptual model, that is, an informal model, in which formulation notions and representations of the subject area knowledge are used. In a broad sense, a conceptual model refers to a meaningful model based on a particular concept or viewpoint.

The conceptual model should be correlated with an enterprise strategy and solve its strategic tasks:

> Enterprise models provide a detailed organizational context relevant and necessary for strategic planning. <...> Enterprise models identify factors that contribute to and hinder strategic changes in an organization at the detailed level, particularly regarding the interaction of business models and information systems. Thus, they provide a promising basis for the methodological support of strategic decision-making processes (Bock, Frank, Bergman & Strecke, 2016).

The consistency of systems and goals of an organization can be enhanced through the use of interoperability models within and outside an organization. Therefore, an enterprise that supports and uses an integrated set of management models can be considered a model-driven organization (MDO) (Clark, Kulkarni, Barn, France, Frank & Turk, 2014).

The choice of a particular model type depends on the type of enterprise activities, tasks it faces, automation of the existing information flows, and resource restrictions. Although the relevance of enterprise modeling is undeniable, a holistic concept of an enterprise model has not yet been developed. New concepts of enterprise modeling are developed depending on the application area and the underlying concept. For example, there is Model-Based Systems Engineering (MBSE) (Dori, 2016) or Model-Based Interoperability Engineering (MoBIE) (Mordecai, Orhof & Dori, 2016).

The main objective of this study is to analyze the existing methodologies for modeling enterprise activities and develop proposals for forming an enterprise information model of the machine-building profile. The solution to this problem is associated with choosing private quality indicators for the software systems and criterion forms of their suitability and optimality.

2 Materials and Methods

Analyzing the existing methodologies and corresponding software systems, which help implement them and build an enterprise conceptual model, is the main research method.

A functionality indicator of a software system is as follows:

$$K_{\langle m\rangle}^{pr} = \left\langle k_1^{pr}, k_2^{pr}, ..., k_m^{pr} \right\rangle,$$

where k_i^{pr} , i – 1, ..., m are private indicators of functionality.

The criteria of functionality include the criteria of suitability (G), optimality (O), and superiority (S).

The authors consider n software systems with m private criteria, where k_{ij}, i = 1, ..., m, j = 1, ..., and n is an indicator of i property of j system; $K_{\langle m\rangle}^{(j)} = \left\langle k_{1j}, k_{2j}, ..., k_{mj} \right\rangle$ is a vector indicator of the functionality of j system; $\left\{k_{ij}^a\right\}$ is a permissible value set of k_{ij} indicator.

The suitability criterion is as follows:

$$G : \bigcap_{i=1}^{m} \left(k_{ij} \in \left\{k_i^a\right\}\right) \oplus U,\ j \in [1, ..., n], \tag{1}$$

where U is a valid event.

The optimality criterion is as follows:

$$O : \bigcap_{i=1}^{m} \left(k_{ij} \in \left\{k_i^a\right\}\right) \bigcup \bigcup_{\forall l \in \{l\}_{m0}} \left(k_{ij} = k_l^{opt}\right) \oplus U,\ j \in [1, ..., n], m_0 \in [1, ..., m], \tag{2}$$

where l is a number of properties to be optimized; m_0 is a quantity of properties to be optimized; $\{l\}_{m_0}$ is a set of properties to be optimized; k_l^{opt} is the optimal value of l property index.

The superiority criterion is as follows:

$$S : \bigcap_{i=1}^{m} \left(k_{ij} \in \left\{k_i^a\right\}\right) \bigcap_{j=1}^{n} \bigcap_{i=1}^{m} \left(k_{il} \geq k_{ij}\right) \oplus U,\ l \in [1, ..., n]. \tag{3}$$

A software system, which condition (3) is met, exceeds other systems in functional characteristics (Mohammadi, 2017). The considered criteria are in the following ratio:

$$S \subset O \subset G.$$

Existing information systems, such as CAD / CAE / CAM / PDM / MES / FRP / MRP / BI / EAM / BRP / CRM / DRP / SCM / HRM / ERP (Computer Aided Design / Computer Aided Engineering / Computer Aided Manufacturing / Product Data Management / Management Execution System / Finance Requirements Planning / Material Requirements Planning / Business Intelligence / Enterprise asset management / Business Resource Planning / Customer Relationship Management / Distribution Requirements Planning / Supply Chain Management / Enterprise Resource Planning / Human Resource Management), focus on specific local applications in most cases.

Fig. 2 shows an integrated enterprise information environment that includes a resource management environment.

A single space of a multidimensional model designed for synchronized data exchange and real-time dynamic enterprise management cannot be built on a single type of simulation system. There is a significant functional and information gap between these systems that is eliminated by various methods. Therefore, the solution to the problem of business process interaction requires using the PLM (Product Lifecycle Management) concept, which provides the possibility of integrating CAD/CAM/CAE/PDM systems into a single information space through structured data files or APIs (Application Programming Interface).

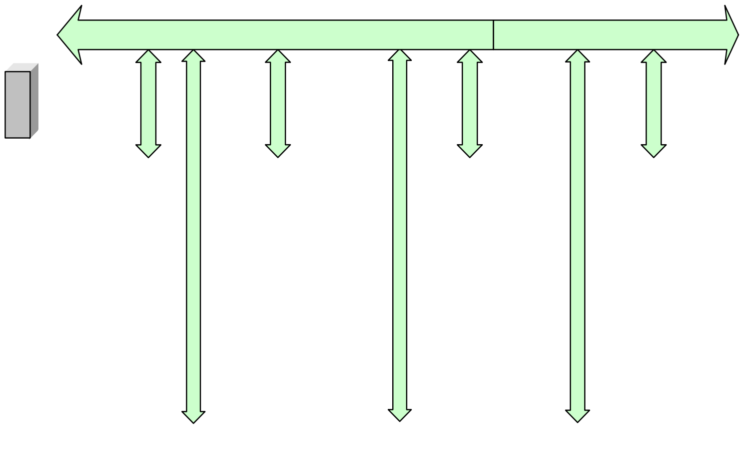

Fig. 2. An integrated enterprise information environment. *Source*: Compiled by the authors.

In the development of enterprise models, structural, hierarchical, behavioral, and other aspects of enterprise activities are considered. Through the structural (architectural) aspect, an enterprise model is considered in the architecture form of an information object. An information model describes enterprise architecture using formal methods (graphs, matrices, lists, tables, and charts). The semantic aspect allows defining and connecting the concepts used in the model to construct elements and link maps (Golenkov, 2011).

The hierarchical aspect allows defining hierarchical abstraction levels of an enterprise model. In this case, objects of the real world and abstraction can be ranked and arranged as a hierarchy of parts determining a model decomposition or a hierarchy of shapes representing the levels of generalization and specialization for building simulated entities.

The behavioral aspect concerns the consideration of functional relationships between system elements and the identification of variables. Behavioral description can be static and dynamic. In static behavior, the model is implemented as a series of object links. Dynamic description uses the information about dynamic characteristics and time dependency of object behavior, attributes, and relationships.

An integration service model, which ensures the functional consistency of information objects in heterogeneous environments, is one of the most important private models. In this case, the integration service definition should refer to standard definitions of information exchange protocols, such as STEP, XML, and EDI.

Private enterprise models include relevant elements of syntax, semantics, and ontology. The model syntax is associated with allowed view relationships. The model semantics extends to objects and links in accordance with a conceptual model of an enterprise (Industrial automation systems – Concepts and rules for enterprise, 1998). A Semantic Unified Meta-Model (SUMM) can serve as an example of a unified model template.

The ontology includes a conceptual apparatus and a thesaurus developed on its basis, and collection and processing information about the system and its constituent objects (Shustova, 2015). At the same time, according to the enterprise model hierarchy, there are application ontologies, subject area and basic subject area ontologies, and ontologies of the upper level.

The modern approach to enterprise modeling is to consider an enterprise from two points of view: (1) as an open set of harmonized business processes aimed to achieve goals and solve the tasks of an enterprise and (2) as the integration of functional entities (software and hardware, personnel, and technologies) (Enterprise integration – Constructs for enterprise modeling, 2007). The latter approach allows an enterprise to be managed as a complete system (Fig. 3). This approach was adopted as a basis of the conceptual model.

Enterprise architecture can be considered as an information base of assets that determine the mission and strategy of an enterprise, the necessary technologies to implement the strategy, and asset adapting processes and technologies for changing needs of the mission (Chen, Doumeingts & Vernadat, 2008). There are several mechanisms for describing architecture, such as the point of view concerning architecture, the architecture description language, and the architecture structure. The combination of enterprise models at the conceptual, block, and element levels is a part of enterprise architecture.

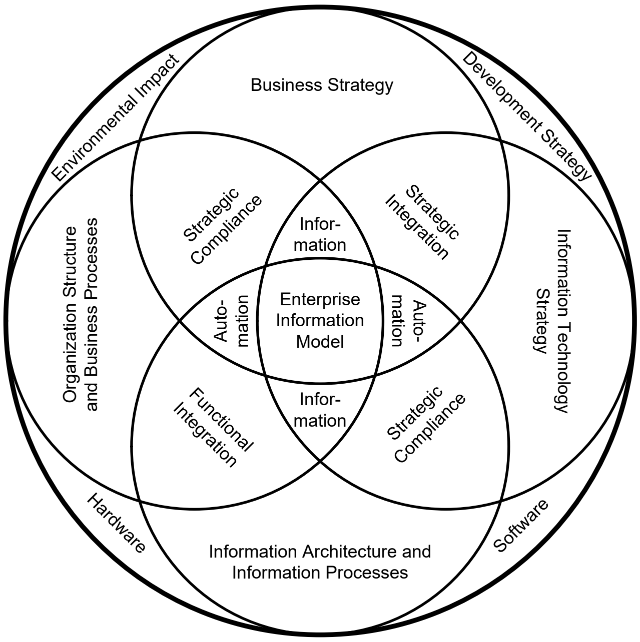

Fig. 3. A structural diagram of an enterprise information model. *Source*: Compiled by the authors.

An enterprise conceptual model is a compact statement of the business essence and its presentation by users and developers within the accepted ontology. The block-level includes reusable reference models based on language modeling designs. An enterprise model structuring at the block level is performed with consideration of various standpoints of the conceptual level.

Research focusing on the development of the enterprise information architecture has led to the creation of various methodologies, such as Zachman Framework, Structured Analysis and Design Technique (SADT), Enterprise Architecture Planning (EAP), Model Driven Architecture (MDA), Enterprise Wide Information Technology Architecture (EWITA), Global Returnable Asset Identifier (GRAI), Integrated Methodology (GRAI-GIM), GRAI Evolution Methodology (GEM), Federal Enterprise Architecture Framework (FEAF), Department of Defense Architecture Framework (DoDAF), Strategic Architecture Model (SAM), Extended Enterprise Architecture Framework (E2AF), Computer Integrated Manufacturing (CIM), Architecture of Integrated Information Systems (ARIS), Computer Integrated Manufacturing Open Systems Architecture (CIMOSA), Technical Architectural Framework Information Management (TAFIM), Technology Neutral Architecture (TNA), Purdue Enterprise Reference

Architecture (PERA), Department of Defense Architecture Framework (DoDAF), The Open Group Architecture Framework (TOGAF), Generalized Enterprise Reference Architecture and Methodology (GERAM), Semantic Object Model (SOM), Dynamic Essential Modeling of Organizations (DEMO), and Method for Multi-perspective Enterprise Modeling (MEMO).

Many modeling methodologies are no longer supported. Limiting their ability to manage certain aspects of the simulated system is the main reason for the decrease in interest in these methodologies.

The enterprise information architecture concept developed to date is a holistic approach that integrates business and information technology capabilities. Most modern enterprise information architectures have four presentation levels: business, application, technology, and data architecture.

Enterprises widely use architecture description languages (ADLs), such as AADL, ARIS, EADL, C2 SADL, ArchiMate, BPMN, SysML, to implement architectures and models. The Unified Modeling Language (UML) is a well-known language for describing enterprise architecture and models (Kaidalova, Seigerroth & Hersson, 2015; Dossou & Pawlewski, 2010). Its modifications, such as UEML, UML4ODP, allow developing the following models:

- Functional models (by creating user-defined class diagrams);
- Object models reflecting the structure and hierarchy of system objects, attributes, operations, and relationships (by implementing object class diagrams);
- Dynamic models reflecting the system behavior (by using dynamics diagrams and process states).

The following requirements are imposed on enterprise models:

- Maintaining the possibility of joint analysis and information system and action system development taking into account the main points of view;
- Providing abstractions corresponding to the professional level of potential users and model representations in the diagram form;
- Comparability of model concepts with implementation level concepts;
- Ensuring the compliance of multilingual diagrams and the existing tool environment;
- Convenient and safe design of enterprise-specific models;
- Planning and ensuring the economic effect of enterprise modeling;
- Availability of enterprise reference models;
- Provision of appropriate modeling tools;
- Availability of convenient specification and enterprise model adaptation.

The authors consider some standard enterprise architectures.

TOGAF (The Open Group, 2009) contains an architecture development method (ADM), which includes eight main stages and links to certain types of charts. However, TOGAF is not suitable as a specification of complex modeling languages that limits its application to implement an enterprise model.

CIMOSA (Amice, 1989) contains a high-level framework that allows the creation of industrial enterprise information systems that include reusable template models but does not include domain-specific modeling languages (DSML). This system is not currently supported.

GERAM (Schmidt, 1998) provides the process of creating enterprise models (EMs). Enterprise architecture defines the universal concepts used in integration projects. These concepts include the life cycle of enterprise systems and products, business process models, modeling languages, and integrated representation of models from various points of view.

ARIS (Scheer, 2000) is designed to model and manage the business processes of industrial enterprises. ARIS does not contain a complex meta-modeling language and uses mainly one type of DSML to model business processes.

SOM (Ferstl & Sinz, 2005) supports the development of information systems and includes business process models and DSML for technological process modeling and object-oriented modeling. SOM is based on a systemic approach.

DEMO (Dietz, 2006) focuses on the language of interaction and individual action. DEMO, built on organizational semiotics, philosophical ontology, and a communicative action theory, provides complex concepts for analyzing communication. Different levels of DEMO abstraction allow the reuse of enterprise models. Unlike other methodologies, DEMO does not take a top-down approach to enterprise analysis. It focuses on individual actions, interaction models, and the role of the language (Frank, 2011). Such an approach leads to a more complex description of individual entity interactions and processes and the construction of a complicated enterprise model.

The comparative assessment of developing methods for enterprise information architecture shows the presence of common properties, for example, high-level structures for various points of view or languages and modeling methods. Nevertheless, since DSML and its methods of use are specific to each approach, there are clear differences between methods, and there is no common meta-modeling language that supports DSML integration.

Enterprise modeling languages not only support the presentation and analysis of business process models. Some of them, such as BPMN, simulate workflow diagrams. However, business process models cover only the narrow focus. The development of information systems that correlate with business and consider strategic options requires the inclusion of more enterprise aspects (Frank, 2012).

Under the creation of an enterprise model, modeling languages are used since they allow the reconstruction of language concepts typical for different points of view of stakeholders. These languages include General Purpose Modeling Languages (GPML), such as UML (Object Management Group, 2019). The disadvantage of this approach is that GPML may require reconstructing concepts, such as *class* and *attribute*, at the domain level.

Different approaches to enterprise conceptual modeling as a tool that helps decrease the model complexity aim primarily to design software systems but not to create an enterprise information model that includes all aspects of its activities.

In this regard, the method for multi-perspective (cognitive) enterprise modeling MEMO and the corresponding meta-modeling environment (Frank, 2011) seems interesting.

Special attention is paid to the analysis of enterprise prospects in this method. MEMO language architecture meets the requirements of adaptability, extensibility, and integration. It consists of the meta-modeling language MEMO MML and the specific DSML. MEMO is characterized by complex language specifications and method engineering support.

MEMO covers the engineering, management, economic, and social aspects of enterprise activities and illustrates current and potential scenarios. An enterprise model includes object and component models integrated with conceptual, strategy, or business process models.

MEMO architecture is based on developing the concept of multidimensional corporate models and extensible language architecture. The appropriate modeling environment integrates editors for different DSML into multilingual model editors. MEMO includes a meta-data editor that enables the user-friendly development, expansion, and use of the supported DSML set and upholds the creation of appropriate graphics model editors (Frank, 2006). The semantics of MEMO meta-models are similar to object-oriented programming languages.

A set of modeling methods and languages used is not always sufficient due to the variety of problems encountered in enterprise modeling. MEMO introduced support for customizing modeling methods, which allows creating fundamentally new enterprise models to cover a broader range of issues.

The object process methodology (OPM) represents a conceptual approach, language, and modeling methodology applied in industrial enterprises (Zdun, Hentrich & Van Der Aalst, 2006). OPM uses two semantically equivalent ways to represent the same model: (1) graphic, implemented using an object process diagram (OPD), and (2) text, consisting of automatically generated program texts in OPL (Object Process Language). OPM allows supporting conceptual modeling with formal syntax and semantics, including objects, processes, information exchange, and system improvement (Mordecai & Dori, 2016), which allows creating an enterprise model not related to a specific application area and including production, organizational, and economic subsystems (Mordecai, Chapman & Dori, 2013).

OPM uses two types of elements: entities and links. Entities are objects, states, and processes. The links used in OPM can be structural and procedural. Structural links form relationships between two objects and are constant. Procedural links depend on the state of objects and processes.

The structural OPM-model of an enterprise is a set of elements connected by structural relationships. The dynamic OPM-model reflects mechanisms of element transformation (Bibliowicz & Dori, 2013). The structural OPM-model of an enterprise integrates functional, structural, and behavioral aspects of objects into a single unified model, which allows coordinating criteria for assessing the correctness of an enterprise model and physical object functioning (Zhou, Kong, Geng, Oiao & Dai, 2019).

The object can be in one of its states or during the transition between them. The process converts objects by creating, using, or changing their state. The processes complement objects by creating a dynamic behavioral aspect of a system. At the same time, OPM does not explicitly define a process state model.

With OPD, it is possible to build a system diagram (SD) of the upper level that provides the context of system target functions. When using OPM, objects with a time-constant state and processes that transform them form the universal minimum ontology.

In order to define DSML, OPM has developed a visual meta-modeling language using a static structural meta-model of the language and a set of rules for checking it. This approach was used to determine a subset of the modeling language using the Eclipse platform and technologies. Thus, OPM allows integrating widespread logical DSML, developed, for example, using EXPRESS.

EXPRESS is a language of information requirement specifications for data structuring, which provides the machine-oriented representation of product data and its exchange. EXPRESS schema is the basis for determining and structuring the interrelation of elements presenting product data.

Along with the widespread EXPRESS language, there are other languages for engineering information modeling designed to solve industrial automation problems, for example, AutomationML. This XML-based language is designed to support the engineering instrumentation data exchange in equipment projection and process management.

Typical object information models include information about the topology, geometry, kinematics, and logic. Simultaneously, logic considers process sequence and order, and behavioral and management aspects of objects.

3 Results

Based on the results of considering concepts and modeling language capabilities for building enterprise information models, the authors have found that MEMO and OPM have the most functional potential.

There are many essential differences between these concepts. For example, MEMO offers its modeling language and its formal model for each service. In OPM, the formalization level is higher, which reduces development costs and increases the reliability of systems and processes. In MEMO, the relationship between objects is presented as a sequence of object actions, while in OPM, it is presented as a sequence of object states that also increases reliability.

The use of the environment for presenting graphical programs, followed by automatic generation of the object code, is the most common component in the methodology description of enterprise information architecture. The authors have compared OPL and UML used as graphical modeling tools.

Each of these languages uses an automatic object code generator to improve performance and program development speed. A way to compare modeling languages was to study the possibility of generating an object code. For this, the authors have used freely available packages of software systems StarUML v1 with a code generator from PostgreSQL and OPCAT with an OPM-GCG code generator. Output results were obtained in XML codes.

The authors have considered graphical representations of UML and OPL programs for providing a milling workstation with a tool as an example and generated object codes. When comparing the UML results, the authors have found the object code incompleteness that fully reveals the structure of objects and only partially their behavior. Some redundancy of the generated code was also detected. The code size generated in OPCAT was 1.3 times less than the code size produced in StarUML.

Various DSML can be used as languages for describing physical objects and processes, including EXPRESS as the most common in engineering enterprises and AutomationML. The latter allows direct interaction with universal languages.

The importance of OPM is gradually increasing, as evidenced by the publication of a preliminary standard PNST 173-2016 (Automation systems and integration – Object-process methodology, 2015) at the object procedure methodology in Russia.

4 Discussion

Implementing a unified enterprise information model is often impossible because it involves too many resources and risks. Nevertheless, it is necessary to develop approaches that determine the evolutionary implementation of an integrated or combined enterprise model at each enterprise.

The construction of a complex information management system for industrial enterprises based on methodologies not discussed in this paper, such as service-oriented architecture (SOA), service-oriented modeling and architecture (SOMA), and Internet of Things (Li, Zhu & Yang, 2015), is complicated due to the lack of their development for use in engineering enterprises in the current conditions.

Systems-oriented design methods do not consider interconnection and collaboration issues. OPM is also intended for integrated modeling and design of different complex dynamic systems and the provision of an interface infrastructure and relationships at the different detail levels. It facilitates a smooth transition from a set of different system-oriented views to a consolidated, integrated model that considers the integration aspects, structure and behavior of the interface and payload, interconnection processes and services, and ultimately new interoperability capabilities (Mordecai & Dori, 2013).

Based on the above, the authors have concluded that OPM semantics is more adapted to the system design than MEMO and other enterprise modeling systems since OPM allows modeling information, hardware, people, and management and creating a complete enterprise information model at the conceptual level.

Although the analysis volume does not prevent fully imagining the advantages and disadvantages of the considered software systems, the authors have assumed that there is no evident superiority between them in most characteristics. However, the authors have also concluded that OPM surpasses other modeling systems in two properties (reliability and program execution speed). In addition, the possibility of using one methodology for developing software, hardware, and information system in general, using one design concept, principles, and environment is an important step towards integrating and optimizing the design of new systems (Ahmad, Bibliowicz, Wengrowicz, Levi & Dori, 2020).

5 Conclusion

The analysis of software tools has shown that there are multidirectional trends in approaches to enterprise model development. The choice of the most promising directions in the sphere of machine-building enterprise modeling is related to the automation level of these enterprises and the presence of certain managing, transmitting, and information processing means, that is, existing restrictions, including financial ones.

Machine-building enterprises in Russia need not separate local management systems but indivisible management systems that consider all enterprise activities and provide the possibility of forecasting, risk management, and optimization. Simultaneously, the development and implementation of information management systems with modeling functions remain a serious problem for many organizations.

Product and enterprise lifecycle management, based on MBSE (Model-Based System Engineering) (Al-Fedaghi & Alnasse, 2018) and the development of network technologies to implement this methodology, is the further development of OPM and integrated approach in enterprise management.

References

Ahmad, J., Bibliowicz, A., Wengrowicz, N., Levi, N., & Dori, D. (2020). Toward integrating systems engineering with software engineering through object-process programming. In M. N. Hoda (Ed.), *International Journal of Information Technology* (pp. 1-35). Cham, Switzerland: Springer. Retrieved from DOI: 10.1007/s41870-020-00488-8

Al-Fedaghi, S., & Alahmad, H. (2017). Orientation in Conceptual Modeling Frameworks. Proceedings of IEEE 2017: *15th International Conference on Dependable, Autonomic and Secure Computing, 2017 IEEE 15th International Conference on Pervasive Intelligence and Computing, 2017 IEEE 3rd International Conference on Big Data Intelligence and Computing and 2017 IEEE Cyber Science and Technology Congress, DASC-PICom-DataCom-CyberSciTech* (pp. 1298-1303). Orlando, FL: Institute of Electrical and Electronics Engineers (IEEE). DOI: 10.1109/DASC-PICom-DataCom-CyberSciTec.2017.209

Al-Fedaghi, S., & Alnasse, H. (2018). Network architecture as a thinking machine. *International Conference on Computational Science and Computational Intelligence (CSCI)* (pp. 884-889). Las Vegas, NV: IEEE. doi:10.1109/CSCI46756.2018.00175

Amice, E. C. (1989). *Open system architecture for CIM*. Berlin. Germany: Springer.

Automation systems and integration – Object-process methodology. (2015). ISO/PAS 19450:2015 from December 5, 2016. Moscow, Russia: Standartinform.

Bibliowicz, A., & Dori, D. (2013). Creating domain-specific modeling languages with OPM/D: A meta-modeling approach. In J. Cordeiro, D. Marca, & M. Sinderen (Eds.). *Proceedings of the 8th International Joint Conference on Software Technologies* (pp. 473-479) (Vol. 1). Reykjavik, Iceland: Springer.

Bock, A., Frank, U., Bergmann, A., & Strecke S**.** (2016). Towards support for strategic decision processes using enterprise models: A critical reconstruction of strategy analysis tools. In J. Horkoff, M. Jeusfeld, & A. Persson (Eds.), *The practice of* *enterprise modeling* (pp 41-56) (Vol. 267). Cham, Switzerland: Springer.

Chen, D., Doumeingts, G., & Vernadat, F. B. (2008). Architectures for enterprise integration and interoperability: Past, present, and future. *Computers in Industry*, *59*, 647-659.

Clark, T., Kulkarni, V., Barn, B., France, R., Frank, U., & Turk, D. (2014). Towards the model-driven organization. In *Hawaii International Conference on System Sciences* (pp. 4817-4826). Waikoloa, HI: IEEE.

Dori, D. (2002). *Object-process. Methodology: A holistic systems approach*. Berlin/Heidelberg, Germany: Springer.

Dori, D. (2016). *Model-based systems engineering with OPM and SysML*. New York, NY: Springer-Verlag.

Dossou, P.-E., & Pawlewski, P. (2010). Using multi-agent system for improving and implementing a new enterprise modeling tool. In Y. Demazeau, Y., F. Dignum, J. M. Corchado, J. Bajo, R. Corchuelo, E. Corchado, F. F. Riverola, V. J. Julián, P. Pawlewski, A. Campbell (Eds.), *Trends in Practical Applications of Agents and Multiagent Systems* (Vol. 71, pp. 225-232). Heidelberg, Germany: Springer.

Enterprise integration – Constructs for enterprise modelling. (2007). HOST R ISO 19440:2007 from December, 2007. Moscow, Russia: Standartinform.

Ferstl, O. K., & Sinz, E. J. (2005). *Modeling of business systems using SOM* (2nd ed.). Berlin, Germany: Springer.

Frank, U. (2006). *Towards a pluralistic conception of research methods in information systems research. Technical Report, 7.* Essen, Germany: ICB, University Duisburg-Essen.

Frank, U. (2011). *MEMO organization modeling language (2): Focus on business processes. ICB-Research Report, 49*. Essen, Germany: University Duisburg-Essen. Retrieved from https://www.econstor.eu/bitstream/10419/70905/1/738565474.pdf

Frank, U. (2011). *Outline of a method for designing domain-specific modeling languages. Technical Report, 42*. Essen, Germany: ICB, University Duisburg-Essen.

Frank, U. (2012). Multi-perspective enterprise modeling: Foundational concepts, prospects, and future research challenges. Proceedings of HICSS: *35th Annual Hawaii International Conference on Systems Sciences* (pp.72-81). Los Alamitos, CA: Computer Society Press.

Golenkov, V. V., & Gulyakina, N. A. (2011) Principles of building mass semantic technology of component design of intelligent systems. In V. V. Golenkov, N. A. Gulyakina (Eds.). Proceedings of OSTIS'2011: *The Open Semantic Technologies for Intelligent Systems*. Minsk, Belarus: Belarusian State University of Informatics and Radioelectronics (BGUIR).

Industrial automation systems – Concepts and rules for enterprise. (1998). ISO 14258:1998 from September 01, 1998. Moscow, Russia: Standartinform.

Kaidalova, J., Seigerroth, U., & Hersson, A. (2015). Towards Guiding the Use of Enterprise Modeling in the Context of Business and IT Alignment. In J. Ralyté, S. España, & Ó. Pastor (Eds.), *The Practice of Enterprise Modeling. PoEM 2015. Lecture Notes in Business Information Processing* (Vol. 235). Cham: Springer. DOI: 10.1007/978-3-319-25897-3_2

Li, Z., Zhu, Y., & Yang, F. (2015). SoS architecture alternatives trade space modeling and computable experimentation: A framework with system engineering thinking. Proceedings of CICN 2015: *International Conference on Computational Intelligence and Communication Networks* (pp. 677-681). Jabalpur, India: Institute of Electrical and Electronics Engineers (IEEE). DOI: 10.1109/CICN.2015.139

Mohammadi, M. (2017). Combination of modeling techniques for business process modeling. *International Journal on Advanced Science, Engineering, and Information*

Technology, *7*(3), 1038-1048.
Mordecai, Y., & Dori, D. (2013). A model-based framework for architecting system-of-systems interoperability, interconnectivity, interfacing, integration, and interaction. *INCOSE International Symposium*, *23*, 1234-1255. DOI: 10.1002/j.2334-5837.2013.tb03083.x
Mordecai, Y., & Dori, D. (2016). Agile modeling of an evolving ballistic missile defense system with Object-Process Methodology. In *2015 Annual IEEE Systems Conference (SysCon) Proceedings* (pp. 839-846). Vancouver, Canada: IEEE. doi:10.1109/SYSCON.2015.7116855
Mordecai, Y., & Dori, D. (2017). Model-based requirements engineering: Architecting for system requirements with stakeholders in mind. Proceedings of *2017 IEEE International Symposium on Systems Engineering* (pp. 1-8). Vienna, Austria: Institute of Electrical and Electronics Engineers (IEEE). DOI: 10.1109/SysEng.2017.8088273
Mordecai, Y., Chapman, C., & Dori, D. (2013). Conceptual modeling semantics for the physical-informatical essence duality problem. In *2013 IEEE International Conference on Systems, Man, and Cybernetics* (pp. 4505-4510). Manchester, UK: IEEE.
Mordecai, Y., Orhof, O., & Dori, D. (2016). Model-based interoperability engineering in systems-of-systems and civil aviation. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, *48*(4), 637-648. DOI: 10.1109/TSMC.2016.2602543
Object Management Group. (2019). *OMG systems modeling language*. Retrieved from http://www.omg.org/spec/SysML/1.6
Scheer, A. W. (2000). *ARIS-business process modeling* (3rd ed.). Berlin, Germany: Springer.
Schmidt, G. (1998). GPN generalized process networks. In P. Bernus, K, Mertins, & G. Schmidt (Eds.). *Handbook on architectures of information systems* (pp. 191-207). Berlin, Heidelberg, Germany: Springer.
Shustova, D. V. (2015). Approach to the development of semantic foundations of information systems for the design and production of aviation equipment. *Ontology of design*, *5*(1(15)), 70-84.
The Open Group. (2009). *The open group architecture framework (TOGAF)* – version 9. Technical report. Retrieved from https://www.up.ac.za/media/shared/Legacy/sitefiles/file/44/1066/2018/8136/tempfolder/togaf9.pdf
Zdun, U., Hentrich, C., & Van Der Aalst, W. M. (2006). A Survey of Patterns for Service-Oriented Architectures. *International Journal of Internet Protocol Technology*, *1*(3), 132–143.
Zhou, B., Kong, D., Geng, H., Qiao, H., & Dai, H. (2019). Complexity modeling of electronic information equipment system of systems based on object-process methodology. *High Power Laser and Particle Beams*, *31*(6), 063202. DOI: 10.11884/HPLPB201931.180328