Complex Systems Informatics and Modeling Quarterly (CSIMQ)
eISSN: 2255-9922
Published online by RTU Press, https://csimq-journals.rtu.lv
Article 196, Issue 36, September/October 2023, Pages I–II
https://doi.org/10.7250/csimq.2023-36.00

Overcoming System Complexity using Models and Knowledge Structures. Editorial Introduction to Issue 36 of CSIMQ

Erika Nazaruka[1*] and Tarmo Robal[2]

[1] Institute of Applied Computer Systems, Riga Technical University, 6A Kipsalas Street, Riga, LV-1048, Latvia
[2] Tallinn University of Technology, 3, Ehitajate tee 5, 19086, Tallinn, Estonia

erika.nazaruka@rtu.lv, tarmo.robal@taltech.ee

Since ancient times to nowadays people have used different means to deal with the complexity of the surrounding environment. These means have helped to minimize the complexity either by decomposing complex things, like systems, into smaller parts or by excluding (abstracting) some aspects that are considered unnecessary for understanding a thing at the moment. Models are one of those means that allow us to simplify reality and give advantages to both decomposition and abstraction. Models can have various forms from textual, tabular, mathematical, and graphical to a combination of these formats. Formal models can be processed, or even executed, by machines. An engineering model must satisfy such characteristics as abstraction, understandability, accuracy, predictiveness, and inexpensiveness [1].

Models explicitly represent knowledge of the modeled domain in a form suitable for reasoning about them and learning. Knowledge may be descriptive, structural, procedural, meta-, or heuristic [2]. Focus on one type of knowledge during the analysis may ignore the other one. Moreover, analysis and reasoning also rely on data representation forms [2], which may lose accuracy due to simplification and different assumptions. Therefore, completeness, correctness, and adequacy of knowledge as well as particularities of the representing structure may affect the results of knowledge processing and decision making [3]. Therefore, the capability of models (and other structures) to represent knowledge completely, adequately, and accurately is still a matter of various research activities. This issue of CSIMQ is devoted to this matter.

The first article, “Design Objectives for Evolvable Knowledge Graphs”, illustrates research results on the representation of human knowledge in the form of knowledge graphs (KGs) that could be used for building artificial intelligence (AI) systems. The authors aim to develop an intelligent assistant to support maintenance personnel in managing a complex cyber-physical system (CPS). This requires the development of an integrated process of KG creation and maintenance that the industry would be able to apply and, thus, to adopt the intelligent assistant to their needs. This article represents the next step in the refinement of this process. The main scientific contribution of the article is related to the better understanding and definition of design objectives for an evolvable KG that should be a core of such AI assistants. A review of existing

* Corresponding author

Reference: E. Nazaruka and T. Robal, “Overcoming System Complexity using Models and Knowledge Structures. Editorial Introduction to Issue 36 of CSIMQ,” Complex Systems Informatics and Modeling Quarterly, CSIMQ, no. 36, pp. I–II, 2023. Available: https://doi.org/10.7250/csimq.2023-36.00

Additional information. Author ORCID iD: E. Nazaruka – https://orcid.org/0000-0002-1731-989X and T Robal – https://orcid.org/0000-0002-7396-8843. PII S225599222300196X. Received: 30 October 2023. Available online: 31 October 2023.

literature on the KG construction and maintenance process and an industrial case study allowed the authors to consolidate and synthesize knowledge and provide a cohesive understanding of the critical elements within the process. This will open broader perspectives on gathering and using industrial knowledge in a disciplined manner.

The second article, “An Actor-oriented and Architecture-driven Approach for Spatially Explicit Agent-based Modeling” represents a joined use of agent-based and spatial modeling for understanding land use and land cover change processes. The main problem in this field is the lack of realistic models that may be rapidly built and maintained. The authors suggest the spatially explicit agent-based modeling process that leverages principles of both agent-based modeling and model-driven development and illustrate its application in a modeling case study of the simulation of hunting and animal population dynamics. The authors prove that the proposed process could increase the use of domain knowledge obtained from various stakeholders (experts) in the domain and automatically generate an initial model code. Thus model validity and re-usability can be improved.

The third article, “Using Fractal Enterprise Modelling in Strategic Analysis with Focus on Intangibles: Empirical Study in Product Innovation” is devoted to modeling and analyzing intangible resources such as knowledge, competencies, learning, capabilities, organizational culture and so on that may also affect enterprise abilities to perform their business functions. The author applies the fractal enterprise model (FEM) as a means for representing interconnections between the intangible and tangible aspects of the organization, i.e. the author proposes meta-constructs that can be instantiated to build models for different processes and extract patterns for analyzed relationships between tangible and intangible resources. The author demonstrates the application of this approach to a research project. The proposed meta-model allows for representing and analyzing how the intangible assets are intertwined with the physical elements in the organizational system, as well as how they affect the behavior of the system, explicitly.

The concluding article, “Exploring Low-Code Development: A Comprehensive Literature Review” summarizes modeling principles implemented in low-code development. Low-code development is a software development approach with minimum manual coding. The main purpose of low-code development is to speed up software development and to help address the shortage of skilled developers. The goal of this article is to analyze the state-of-the-art in this field. The review of literature is done for the period from 2019 to 2022 by analyzing 42 unique research papers. The article gives answers to questions about the current understanding and applications of low-code development, as well as its available features and particularities. The results of this article show trends in the current application of low-code development and could be used for its further evolution.

The CSIMQ editorial team would like to thank the reviewers for providing their valuable feedback for the submitted articles and all the authors for submitting the articles and reporting their scientifically innovative and passionate work.

References

[1] B. Selic, “The pragmatics of model-driven development,” *IEEE Software*, vol. 20, no. 5, pp. 19-25, 2003. Available: https://doi.org/10.1109/MS.2003.1231146

[2] I. H. Sarker, “AI-Based Modeling: Techniques, Applications and Research Issues Towards Automation, Intelligent and Smart Systems,” *SN Computer Science*, no. 3, article no. 158, 2022. Available: https://doi.org/10.1007/s42979-022-01043-x

[3] P. J. Driscoll, G. S. Parnell, and D. L. Henderson, *Decision Making in Systems Engineering and Management*. 3rd Edition, Wiley, 2022.