Towards an Ontology-Based Platform for Integrating Infectious Disease Simulation Models

Papa Alioune Cisse(✉)

Assane Seck University of Ziguinchor (UASZ), BP 523, Ziguinchor, Senegal
papaaliounecisse@yahoo.fr

Abstract. The control of infectious diseases is a perpetual challenge in public health. The need to understand their complex dynamics and the imperative to develop epidemiological surveillance, control and mitigation strategies often require modeling and numerical simulation activities. However, it should be noted that modeling and simulation of such complex systems requires expert skills in modeling tools and simulation platforms. This is often a stumbling block for practitioners and beginners in the field. In addition, communication between modelers and domain experts can be complicated by differences in the languages and concepts used on both sides. In this paper, we present an ongoing project aimed at providing solutions to these complications in order to facilitate the infectious disease modeling exercise, simplify the model simulation process, foster collaboration between researchers (practitioners and "modelers") and enable reproducibility and reuse of models and model parts. To implement this project, we are setting up a web-based platform for the integration of infectious disease simulation models. The microservices architecture we have proposed for this platform includes a simulation model repository, an ontology for annotating these models, and a set of services enabling autonomous orchestration of simulations by ensuring the selection, comparison, possible composition, formulation and simulation of models in dedicated simulators.

Keywords: infectious diseases · modeling · numerical simulation · ontologies · epidemiological monitoring systems · ontology driven simulation (ODS) · computational epidemiology · microservices architecture (MSA)

1 Introduction

The spread of infectious diseases involves a very large number of entities (hosts, transmission vectors, pathogens, risk factors, etc.) whose interactions give rise to the emergence of epidemiological situations that can occur at different spatial (regional, continental, worldwide) and temporal (seasonal, for example) scales. These types of phenomena, with their evolutions and emergences resulting from the interactions of their component parts, are described as complex systems [1, 2]. To analyze the evolution and dynamics of such complex systems, it is often necessary to resort to systemic modeling [3], which consists in building a model that reproduces its behavior in a simulation.

K. Cheikh M. F. Kebe et al. (Eds.): InterSol 2024, LNICST 610, pp. 264–274, 2025.
https://doi.org/10.1007/978-3-031-86493-3_21

However, modeling and simulating complex systems in general, and infectious diseases in particular, requires expert knowledge of modeling tools (mathematical and/or computer science) on the one hand, and simulation platforms on the other. This is sometimes a major obstacle for practitioners (epidemiologists, biologists, etc.).

On the other hand, to model a disease or part of a disease, the computer scientist or mathematician needs expert knowledge and consistent data on the disease. These data are sometimes very difficult to find among practitioners. Besides, even if data are available, communication between the modelers and the domain experts can sometimes be difficult, due to differences in the concepts, terms and tools used on both sides, and the lack of a common framework for sharing and exchange.

What's more, for a novice modeler, the exercise of learning and self-training about disease modeling can be complicated by a lack of references and examples of models to draw on.

Furthermore, in scientific publications, authors often make their models available to the community, sometimes even the source codes of these models and the data they use. However, it has been found that reproducing the results presented in the publications is difficult, if not impossible, because certain details relating to simulator versions and dependencies, required operating systems and computing environments are omitted [4, 5].

We have recently initiated a project to help solve these problems. In this project, we are implementing a platform for the integration of infectious disease simulation models and simulators. The platform includes a directory of simulation models and simulators of these models, an ontology for annotating models and simulators, and an autonomous simulation orchestration module for the selection, comparison, composition, formulation and simulation of models. In this article, we present the foundations of this ongoing project, the microservices architecture of the platform and its various services.

This document is divided into three sections. The first section presents a few definitions of essential concepts to facilitate the reading of the document, and a bibliographical review of works that have addressed the issues we raise. The second section presents the foundations of our project, its objectives and expected results. The third section discusses the microservices architecture and the various components of our platform.

2 Literature Review

2.1 Definitions of Essential Notions

Complex System. The global behavior of a system, made up of multiple and possibly heterogeneous entities, can be explained by the result of interactions between its constituent entities. It is the ability to determine the system's overall behavior that enables us to characterize it as simple, and if not, as complex [6]. A complex system does not reveal a latent determinism that can be calculated, but manifests a certain form of possible unpredictability and plausible emergence of the new [3]. Indeed, the entities that make up a complex system are endowed with behaviors and sometimes decision-making mechanisms that guide their behavior independently of the other entities. It is the interaction and dynamics of interaction between these entities that determine the overall behavior of the system. However, the unpredictability of complex systems does not rule out their intelligibility. Since it is impossible to represent them definitively, i.e., to produce finite representations that are statically ready for use, we can, at a given moment, make representations that are themselves complex, so as to enable reasoning [7]. This process of producing a simplified representation of a system to make it intelligible is called modeling, and its result is a model.

Modeling. Modeling is the activity of building models. Along with experimentation (or "simulation"), it is one of the two main components of the scientific approach [6]. Indeed, modeling and simulation are two activities that go hand in hand, and together form a fully-fledged scientific approach whose aim is to propose specific theories, tools and vocabularies in order to produce knowledge about natural and artificial phenomena [8]. Modeling consists in constructing an abstraction of the system or phenomenon under study (the model), which retains only the characteristic quantities (state variables) of the system deemed relevant by the modeler [9].

Model. In science, a model is a simplified representation of a portion of reality from a particular point of view or in response to a particular question. There are two families of models: "physical" models and "abstract" models. Physical models (including scale models and animal models) are physical devices designed for experimentation. The latter are models designed to be implemented and run/simulated on a computer [6]. It is this second category of models that interests us in the context of this work.

The main purpose of a model is to be more explicit, simpler and easier to manipulate than the reality it is supposed to represent. To this end, models ignore a large number of details about the reality they represent, in order to focus on data deemed more relevant to the problem under study. In this way, there is a homomorphism between the object of study and the model that represents it. This makes it possible to apply the results of the model to the object itself [7], as shown in the following Fig. 1.

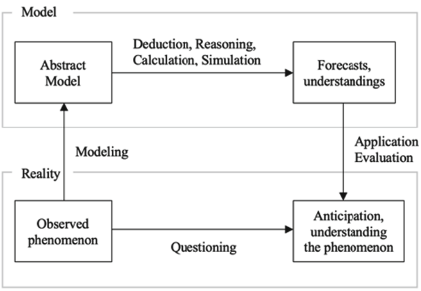

Fig. 1. Forecasting and understanding phenomena requires the development of models [7].

The observed phenomenon is translated into an abstraction (the model). This abstraction is then manipulated (by simulation, for example) to obtain results which can then be used to better understand or predict future situations of the observed phenomenon. Mechanisms of this kind can be used to predict the evolution of a disease in a given population, or to evaluate policies for controlling the spread of a disease or epidemic (vaccines, treatments, etc.).

Static vs. Dynamic Model. A model is called static when its purpose is to represent the structure of a reference system photographed at a given moment, without any reference to its evolution over time. Conversely, a model will be called dynamic when it includes in its representation assumptions or rules concerning the evolution over time of the reference system [6]. It should therefore be remembered that only dynamic models can be subjected to a process of experimentation or simulation. A static model, by definition, cannot be simulated.

Simulation. Simulation is the activity by which, according to precise objectives, and with the help of a computerized experimental device (called a simulator), a dynamic model is disturbed according to a determined protocol, making its inputs evolve and recovering its outputs [6].

Simulator or Simulation Platform. A simulation platform (or simulator for short) is a computer program that simulates a real phenomenon on a computer. In other words, a simulator enables the virtual reproduction of an environment or process.

Model Input/Output Parameters. The inputs of a dynamic model are parameters whose values are defined outside the model and which represent what the simulator can perturb. The outputs of a dynamic model are also parameters that express what we want to measure in response to these disturbances [6].

Ontology. Ontologies are interested in representing the knowledge of a domain. They consist more precisely in the identification of concepts and their relations, and in proposing their formal representation for resource annotation and semantic reasoning. Based on these representations, an autonomous agent will have the ability to understand and effectively manage a system, reason and make deliberations [10].

2.2 Related Works

In the literature, we have found a large number of works in line with the study we are proposing in this project, particularly in the areas of infectious disease model warehousing, sharing, reuse and reproducibility of models:

Models of Infectious Disease Agent Study (MIDAS)[1]. MIDAS is "a global network of scientists and practitioners who develop and use computational, statistical and mathematical models to improve understanding of infectious disease dynamics". MIDAS provides an organized collection of resources (data and software) relevant to the field of infectious disease modeling in a catalog form[2] to enable searches for specific resources on a disease, topic, location or temporal coverage. In 2008, the MREP (Model REPository) [11] tool was developed under the auspices of MIDAS to catalog models, model run results and store them in a relational database for future use and referencing.

World Health Organization's Disease Outbreak News (WHO's DON). DON is an online news service [12] through which WHO informs the international community about

[1] https://midasnetwork.us/.
[2] https://midasnetwork.us/catalog/.

disease outbreaks. It is the only official online registry of epidemic reports, maintained by WHO since 1996. In [13], the authors noted some limitations of this tool in terms of its analytical utility, which "is severely hampered by its unstructured, text-based system". To make the DON more operational and improve its performance, the authors developed a metadata "skeleton" that stores key information about DON reports (the identity of the report itself, the disease referred to in the report, the geographical area, the raw epidemiological data, the time of epidemic progression and any additional information) in order to constitute a fully-fledged data warehouse on disease epidemics.

Framework for Infectious Disease Analysis (FIDA). FIDA is a software environment and conceptual architecture for the integration of data and knowledge on infectious diseases, visualization, prediction and evaluation of control interventions [14]. To this end, it ensures:

- Automatic and autonomous collection of biomonitoring data (from various sources such as social media, news feeds, websites, etc.) using automatic natural language processing (NLP);
- The integration of structured and unstructured data on the history of disease emergence, endemic strains, environmental conditions, wildlife populations, land-use policies, human habitation and culture, local health infrastructures, etc.;
- The application of advanced machine learning for prediction;
- Multi-modeling, with the integration of several modeling approaches (mathematical, computational, compartmental, agent-based, etc.) to meet different requirements for analyzing the dynamics of infectious diseases.

In [5], the issue of the reproducibility of results of computer models of infectious diseases is addressed. Indeed, in publications relating to infectious disease models, even if input data and model source codes are available, it is notoriously difficult, if not impossible, to reproduce study results [15], because "other essential details such as information on software versions and dependencies, or on the required operating system and computing environments, are often missing" [4]. To address these limitations, [5] proposes "An implementation framework to improve the transparency and reproducibility of computational models of infectious diseases", which can be used by scientific communities to develop usable tools for sharing computer models in a reproducible way.

3 Presentation of Our Project

3.1 Theoretical Background

Our works focuses on three themes:

- **Epidemiological monitoring systems** are used to control the spread of disease by proposing action plans to prevent identified risks [1]. The advantage of an epidemiological monitoring system is that it enables and facilitates risk prediction and decision-making based on quantitative analyses, carried out on the basis of numerical model simulations. These models, built from epidemiological studies, help to explain the dynamics of disease propagation and validate hypotheses.

- **Computational epidemiology** is a discipline whose main objective is the application of computational (including modeling and simulation techniques, approaches and tools) and geographic (including tools for representing and visualizing complex geospatial data) concepts and resources to provide epidemiologists with user-friendly tools, to enable them to better understand the fundamental problems of epidemiology, such as the spread of disease, the effectiveness of a public health intervention, the prediction and analysis of disease manifestations and their spread in a given population, [2, 16].
- **Ontology Driven Simulation (ODS)** is a process that uses the knowledge encoded in ontologies to dynamically and automatically design simulation models. This involves having domain or application ontologies on the one hand, and modeling ontologies on the other (encoding modeling information such as model components, different modeling phases and activities, etc.). Next, domain ontology concepts are mapped to modeling ontology concepts, and modeling ontology instances are created to represent a model. Once the ontology instances representing the model have been created, additional tools are used to translate them into executable simulation models [10].

The platform we present in this work has its roots in these three themes. Indeed, it is an ontology-based platform for integrating infectious disease simulation models, which autonomously orchestrates the simulation process by guiding the selection, comparison and eventual composition (also called coupling) of models.

In terms of epidemiological monitoring systems, the platform offers the services of a numerical model simulation engine with a large “warehouse” of infectious disease simulation models. It is invoked, by providing a “simulation request”, by a human user with numerical data or queries on a particular disease, or by any epidemiological monitoring system collecting field epidemiological data.

In terms of “Computational epidemiology”, the platform aims to integrate, as far as possible, recent solutions resulting from advances in computer science research to meet the new contemporary epidemiological challenges of modeling and simulating the spread of infectious diseases. These include:

- The use of Multi-Agent Systems (MAS) as an additional modeling and simulation approach to capture the complexity inherent in the spread of infectious diseases, which is partly involved in human interactions and behaviors that are apprehended through social and spatial interaction networks.
- The use of new graphic representation and visualization techniques (e.g., Geographic Information Systems and virtual reality) for simulation input and output data.

In terms of ODS, the platform integrates a domain ontology for the semantic annotation of simulation models, data and resources. Based on this ontology and the simulation model repository, the platform orchestrates the simulation process by guiding the comparison, selection and composition of simulation models in response to a “simulation request”.

3.2 Objectives

The overall aim of the project is to propose an ontology-based platform for integrating infectious disease simulation models and automating the simulation process of these models. This general objective is broken down into three specific objectives:

– **Objective 1.** Set up a library (warehouse) of infectious disease simulation models.
– **Objective 2.** Build a domain ontology of infectious disease simulation models to annotate the models in the warehouse.
– **Objective 3.** Implement a platform for model integration, orchestration and simulation automation.

3.3 Expected Results

Among other things, the platform will make it possible to:

– **Run simulations.** Simulate a single model at a time or by model composition (several models and several simulators).
– **Facilitating collaboration.** The platform will enable interoperability of simulation tools, model reuse and data sharing between researchers in the field of infectious disease modeling. This will enhance collaboration opportunities.
– **Serve as research support.** Articles published around infectious disease simulation models can be linked to the ontology to help researchers find more relevant research articles more quickly.
– **Share models and simulation model components.** By providing an ontology, an infrastructure is created for storing and retrieving executable simulation model components. This will facilitate the modeling exercise.
– **Reproduce the results of model simulations.** By hosting dedicated simulators and setting up the conditions for their execution, the platform should enable a model to be simulated under the conditions in which it has been presented in a scientific publication.

4 Microservices Architecture of Our Platform

4.1 Platform Architecture

We propose a microservices architecture for our platform. Definitions, advantages and challenges of a microservices architecture for applications such as the one we propose can be found in these articles [17–20] (Fig. 2).

This is a "data-driven" microservices architecture similar to those presented in [21] and [22]. In [21], knowledge extraction techniques such as data mining, classification and visualization are used to generate graphs enabling actors in a network of researchers to discover the thematic relationships between the various existing publications. A data-driven microservices architecture is proposed. It takes a publication as input and returns a graph containing all publications in the research network having themes in common with this publication. In [22], from a set of microservices that share common topics (functionality, input object, parameters, etc.), the system returns to the user the list of microservices that correspond to the topics requested in his query.

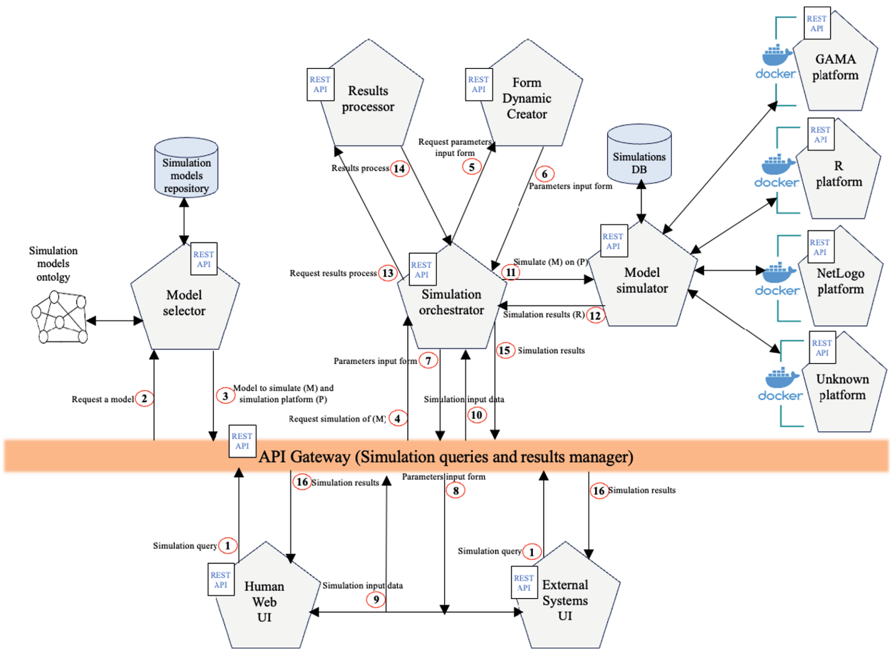

Fig. 2. Workflow for fulfilling a "simulation request" (first version).

The idea of publication discovery in [21] and microservices in [22] is similar to our approach which, starting from what we call a "simulation query", must select, in the best case, the appropriate simulation model to satisfy the query and, in the least, a list of simulation models to compose. In the worst case, if no simulation model can satisfy the user's query, the system must guide the user to formulate one. Once a simulation model has been found, the system should proceed to simulate it with the appropriate simulator and return the simulation results to the user.

4.2 The Different Services of Our Platform

This architecture[3] is generally composed of a "model selection" service, a "Gateway API" support service, user interfaces, a "simulation manager" module comprising a simulation "orchestrator", a "model simulator" service, a "dynamic form creator" service and a "simulation results exploitation" service.

User Interfaces. In our architecture, "user interfaces" are also microservices. They are divided into 2 parts: interfaces consisting of web pages for human users, and interfaces for external systems. These enable users (human or software) to interact with the assistance module using predefined functionalities via "simulation requests" (1). User requests can be accompanied by model simulation input data (9), which will always be supplied as

[3] This first version of the architecture does not yet take into account aspects related to model composition.

input via these interfaces. Simulation results are returned to users via these interfaces too (16).

API Gateway. The API Gateway, called "Simulation queries and results manager", is the helpdesk that handles user queries and simulation results. It receives requests from user interfaces and queries the "model selector" service (2) to determine the model to be simulated and the dedicated simulator. It then queries the simulation orchestrator, passing it the model to be simulated and the simulator (4).

The "Model Selector" Service. This service is based on a "Simulation models repository" and a "Simulation models ontology" for infectious diseases. Based on a user query, it identifies and selects the simulation model to be implemented to satisfy the user's request (3).

Simulation Orchestrator. The simulation orchestrator plays the role of conductor, "initializing" an object representing a simulation, and orchestrating the running and use of simulation results. It takes as input a model to be simulated, a simulation platform and initializes the simulation. If the model to be simulated requires input parameter data, the simulation orchestrator queries the "Form dynamic creator" service (5) to dynamically create a form enabling the user to give values to the model's input parameters (steps 6 to 10). These simulation input values, the model to be simulated, the simulation platform and the initialized "simulation" are then supplied to the "model simulator" service (11) to run the simulations. After simulations (12), the "simulation" object is saved in a database ("Simulation DB"). It is then supplied to the "Results processor" service (13) to define how the simulation results are to be used. These results are then retrieved and returned to the user (14) via the API Gateway (15).

The "Form Dynamic Creator" Service. The role of the "Form dynamic creator" service is to dynamically create forms according to user requirements. It can be used to create forms for simulation input parameters and forms for simulation output evaluation. Indeed, when a model is to be simulated with values to be supplied by the user, a form is dynamically created according to the model's input parameters. To exploit the raw data output from a simulation (which can sometimes be numerous and complex), a form can be created to allow the user to filter the data according to certain criteria in line with the model's simulation output.

The "Model Simulator" Service. It enables the simulation of a model in a dedicated simulator with input data supplied by the user. It is based on a set of dedicated simulators with their different versions. Each simulator is a microservice in a Docker container.

The "Results Processor" Service. It allows the user to exploit simulation results, for example, by visualizing simulation data via a user interface. Data can be loaded in their raw state or visualized with graphs.

5 Conclusion and Perspectives

In this article, we present an ongoing project that aims to make a contribution to facilitating the infectious disease modeling exercise for beginners in the field and non-experts, simplifying the model discovery and simulation process for practitioners, fostering collaboration between researchers (practitioners and "modelers") and in model reproducibility and reuse. We also presented the microservices architecture of a platform currently being implemented to implement this project. This architecture comprises a simulation model repository, an ontology for annotating these models, and a set of services for autonomously orchestrating simulations in dedicated simulators.

Our future works will focus on the details of each of the components of the architecture presented, notably the infectious disease simulation model ontology, the simulation model repository and the autonomous model simulation orchestration services. They will also focus on enhancing our microservices architecture to take into account aspects related to model composition and simulation platform integration.

References

1. Camara, G.: Conception d'un système de veille épidémiologique à base d'ontologies: application à la schistosomiase au Sénégal. Thèse de doctorat, Paris (1971–2017). Université Pierre et Marie Curie, France, Sénégal (2013)
2. Cissé, P.A.: Simulation à base d'agents de la propagation de la Schistosomiase : une approche de composition et de déploiement de modèles. P.hd. thesis, Université Pierre et Marie Curie - Paris VI (2016)
3. Moigne, J.-L.: La modélisation des systèmes complexes. Dunod, Paris (1990)
4. Stodden, V., et al.: Enhancing reproducibility for computational methods. Sci. **354**, 1240–1241 (2016). https://doi.org/10.1126/science.aah6168
5. Pokutnaya, D., Childers, B., Arcury-Quandt, A.E., Hochheiser, H., Van Panhuis, W.G.: An implementation framework to improve the transparency and reproducibility of computational models of infectious diseases. PLOS Comput. Biol. **19**, e1010856 (2023). https://doi.org/10.1371/journal.pcbi.1010856
6. Treuil, J.-P., Drogoul, A., Zucker, J.-D.: Modélisation et simulation à base d'agents: exemples commentés, outils informatiques et questions théoriques. Dunod, Paris (2008)
7. Ferber, J.: Les Systèmes multi-agents: vers une intelligence collective. InterEditions (1995)
8. Siebert, J.: Approche multi-agent pour la multi-modélisation et le couplage de simulations. Application à l'étude des influences entre le fonctionnement des réseaux ambiants et le comportement de leurs utilisateurs. Ph.d., thesis, Université Henri Poincaré - Nancy I (2011)
9. Fianyo, Y.E.: Couplage de modèles à l'aide d'agents: le système OSIRIS. Paris 9 (2001)
10. Cisse, P.A., Camara, G., Dembele, J.M., Lo, M.: An ontological model for the annotation of infectious disease simulation models. In: Bassioni, G., Kebe, C.M.F., Gueye, A., Ndiaye, A. (eds.) Innovations and Interdisciplinary Solutions for Underserved Areas. LNICS, Social Informatics and Telecommunications Engineering, pp. 82–91. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-34863-2_8
11. Cooley, P.C., et al.: The model repository of the Models of Infectious Disease Agent Study. RTI International (2008). http://www.rti.org. https://doi.org/10.1109/TITB.2007.910354
12. Carlson, C.J.: cghss/dons: Preprint version of the data (version v0.1). Zenodo (2022). https://doi.org/10.5281/ZENODO.6374495
13. Carlson, C.J., et al.: The World Health Organization's disease outbreak news: a retrospective database. PLOS Global Public Health **3**, e0001083 (2023). https://doi.org/10.1371/journal.pgph.0001083
14. Erraguntla, M., Zapletal, J., Lawley, M.: Framework for infectious disease analysis: a comprehensive and integrative multi-modeling approach to disease prediction and management. Health Inform. J. **25**, 1170–1187 (2019). https://doi.org/10.1177/1460458217747112

15. National Academies of Sciences, Engineering, and Medicine; Policy and Global Affairs; Committee on Science, Engineering, Medicine, and Public Policy; Board on Research Data and Information; Division on Engineering and Physical Sciences; Committee on Applied and Theoretical Statistics; Board on Mathematical Sciences and Analytics; Division on Earth and Life Studies; Nuclear and Radiation Studies Board; Division of Behavioral and Social Sciences and Education; Committee on National Statistics; Board on Behavioral, Cognitive, and Sensory Sciences; Committee on Reproducibility and Replicability in Science (2019). Reproducibility and Replicability in Science. Washington (DC): National Academies Press (US)
16. Cisse, P.A., Dembele, J.M., Lo, M., Cambier, C.: Multi-agent systems for epidemiology: example of an agent-based simulation platform for schistosomiasis. In: Bajo, J., et al. (eds.) Highlights of Practical Applications of Cyber-Physical Multi-agent Systems. CCIS, pp. 157–168. Springer, Cham (2017)
17. Viggiato, M., Terra, R., Rocha, H., Valente, M.T., Figueiredo, E.: Microservices in Practice: A Survey Study (2018). arXiv. https://doi.org/10.48550/arXiv.1808.04836
18. Dragoni, N., Lanese, I., Larsen, S.T., Mazzara, M., Mustafin, R., Safina, L.: Microservices: How To Make Your Application Scale (2017). arXiv. https://doi.org/10.48550/arXiv.1702.07149
19. Dragoni, N., et al.: Microservices: yesterday, today, and tomorrow (2017). arXiv. https://doi.org/10.48550/arXiv.1606.04036
20. Hassan, S., Bahsoon, R., Kazman, R.: Microservice Transition and its Granularity Problem: A Systematic Mapping Study (2019). arXiv. https://doi.org/10.48550/arXiv.1903.11665
21. Thiele, T., Sommer, T., Stiehm, S., Jeschke, S., Richert, A.: Exploring research networks with data science: a data-driven microservice architecture for synergy detection. In: 2016 IEEE 4th International Conference on Future Internet of Things and Cloud Workshops (FiCloudW), pp. 246–251 (2016). https://doi.org/10.1109/W-FiCloud.2016.58
22. Caron, E., Houmani, Z.: Architecture Microservices pilotée par les données 1 (2018)