From a prototype to a data ecosystem for experimental data and predictive models

Edoardo Ramalli[1,*], Barbara Pernici[1]

[1]Politecnico di Milano, Via Giuseppe Ponzio, 34/5, Milan, 20133, Italy

Abstract

Data ecosystems have been a game-changer in many industrial applications and research fields, speeding up their development. The possibility of collecting large amounts of data within the same environment has also raised some common questions to all application domains, including the quality of the data collected and their reliability and trustworthiness. From experience gained collaborating with the chemical engineering field, this paper raises some discussion points related to the management of experimental data and predictive models within a data ecosystem. In fact, this type of data poses new requirements that require specific treatment before being implemented in a traditional data ecosystem.

1. Introduction

Data ecosystems (DE), in the last years, have shown their potential in boosting the research and the industry, holding a central role in many definitions of industry 4.0 [1]. DEs facilitates and encourages data sharing while extracting knowledge and enhancing the comprehension of a phenomenon [2]. In some cases, the data management features of a DE are even fundamental and a prerequisite to applying data science in a big data context [3]. In addition, DEs lend themselves well to the ongoing scholarly trends of data reuse [4]. In any case, DEs raise many challenges that need to be addressed and tailored based on the domain [5].

A possible application of such an information system is to use it as a collection of tools, scientific repositories, and services to improve the development process of predictive models for physical-chemical phenomena. The development of these data-driven models relies on a manually managed data set. A model computes simulated data (or simulations) that are then manually validated against the corresponding experimental data (or experiments). A DE in this field represents a possible game-changer for several reasons.

First, the number of available experimental data is tiny when compared to other data-intensive application, even if is growing in the last years. Experiments are expensive and time-consuming, while running simulations are computationally heavy. Therefore, sharing and reusing data is a primary objective of the scientific community and one of the principal purposes of employing a data ecosystem in this domain. As in many data-driven applications, "you are what you eat," and concepts such as data quality [6] or database diversity tools [7] are fundamental to building reliable predictive models. Data quality has been proven that has a direct impact on decision-making activities [8], while database diversity could also have relevant social implications in some domains due to the bias presented in the dataset [9]. DEs are protagonists also in other aspects: making data and services converge in the same system can help increase their use and trustworthiness. More data are collected inside a DE, and more users are attracted, whom themselves bring more data. The more active users are in DE, the more the data and services are checked and used, and the more reliable the data and the overall system are. Therefore, having data and tools in the same system is a positive vicious circle, even if starting could be very challenging.

This work presents the experience in designing and implementing a data ecosystem to enhance the development process of predictive models in the field of chemical engineering. This DE needs to manage predictive models, analysis results, experimental and simulated data to extract insights automatically while trying to address the typical challenges a data ecosystem faces during its design, such as transparency and trustworthiness [10].

The need for DEs for storing experimental data and tools in the chemical engineering domain has emerged in the last few years. First attempts to integrate data together with analysis tools were made over time in the PriMe repository [11], where some tools were provided in addition to data, and the need for being able to analyze the data production process and quality of data first emerged. Other repositories storing both experimental data and tools also include systems such as ChemKED[1] or ReSpeCTh [12]. However, there is a lack to support for an approach in the design of simulation models as a process involving all the phases, from experimental data collection to simulation results analysis. This limitation has brought or the abandonment or the creation

Proc. of the First International Workshop on Data Ecosystems (DEco'22), September 5, 2022, Sydney, Australia

edoardo.ramalli@polimi.it (E. Ramalli);
barbara.pernici@polimi.it (B. Pernici)

[1]http://www.chemked.com/

of many alternative frameworks or software focusing on specific aspects (e.g., CloudFlame[2] for flames data and simulations) that are challenging to work together.

This paper discusses the emerging directions derived from the design and use of a prototype system for such purposes. Even if the features of a DE are well defined, implementing and tailoring them in a particular domain and application has its unique challenges. For instance, scientific repositories have well-known problems with data quality [13]. The biggest challenge concerns the design method for our data ecosystem. A top-down strategy requires much time in the design phase, and often consumers are not willing to wait, even if it is the best approach to saving time readjusting or adding new features. On the other side, a bottom-up approach allowed us to deliver a product faster, even if several iterations of feedback-adjustment were required. Nevertheless, this procedure highlighted some requirements that would hardly have emerged with a top-down approach, given the complexity of the application domain.

In any case, four phases were primarily identified during this project, as shown in Figure 1. In each phase, even if some features are not immediately needed in the current product delivery, some design decisions were made keeping in mind the final goal of delivering a data ecosystem. Therefore, this paper presents the challenges and design decisions in each phase toward developing a data ecosystem for a specific application domain.

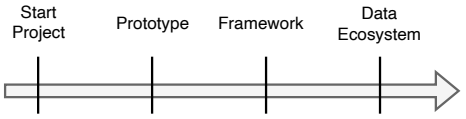

Figure 1: The four stages of our project in the development of a DE in the chemical engineering field.

After the kick-off of the project and the requirement collection, it was delivered the first prototype [14] in which the main characteristics are the creation of a repository, with the proper database schema to collect the data, the architectural structures of various components of the system together with the technological choices.

The second phase regards the framework creation [15]. If the primary purpose of the prototype phase is to collect feedback from the end-user, with the framework, the need was to deliver a product that can be used daily by a single research group. This requirement implicitly suggests that a series of features are needed to ensure good data quality of the database, fault-tolerant features, usability, accessibility, authentication, interoperability, and so on.

[2]https://cloudflame.kaust.edu.sa/

Finally, in the last stage of the project, it was removed the constraint relaxation about the fact that only a small number of people will use the framework, all belonging to the same research group, and de-facto transforming our framework into a data ecosystem.

The paper is structured as follows. In Section 2, the prototype stage of our process is introduced, also presenting the main types of data that will be stored in the DE. In Section 3, it is illustrated the framework version of the project, where design and implementation choices are made to fulfill the typical characteristics of DE. Section 4 shows the challenges and consequences of implementing a DE considering intellectual property data in a collaborative environment. Finally, the data ecosystem's open challenges and future developments are discussed in Section 5.

2. Prototype

In the first phase of the project, the requirements were gathered and discussed continuously with the domain experts (our stakeholders). At the end of the requirements collection phase, it is essential to design properly the architecture and the technology necessary to implement an information system suitable to meet the discussed needs. The resulting product of this phase is a simple prototype to check if the initial requirements are fulfilled and collect new ones. However, it is already necessary to structure the system to be compatible with the final architecture of a data ecosystem, even if some of these features are not strictly necessary for this step. A detailed description of the design decisions in this phase is reported in [14].

In a DE for the development of predictive models, it is a game-changer to gather together experimental data, models, and analysis tools in the same system. These entities define what type of data the final DE should manage: experimental data (experiment), simulated data (simulation), models, and, eventually, analysis results.

From an architectural and implementation perspective, to guarantee maintainability and extensibility over time, it is preferred to choose a micro-service architecture that provides a few simple services, together with a relational database to store experiments, models, simulations, and analyses. Then the user can request and combine them as preferred through an HTTP API, hence separating the front-end from the back-end.

Experimental Data Experiments are actual experimental measurements about the investigation of a particular environmental condition. An experiment is, in fact, correlated with other metadata that characterize, for example, the experiment author, the methodology, and the experimental conditions. These metadata contain a series of information essential to classify the experiments

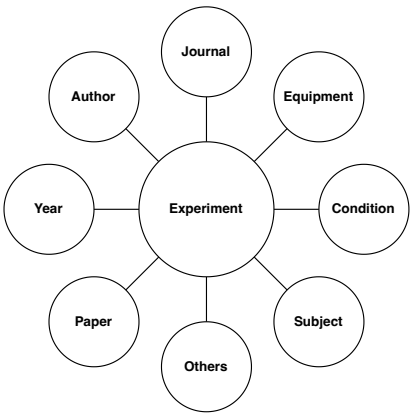

Figure 2: Experimental data metamodel

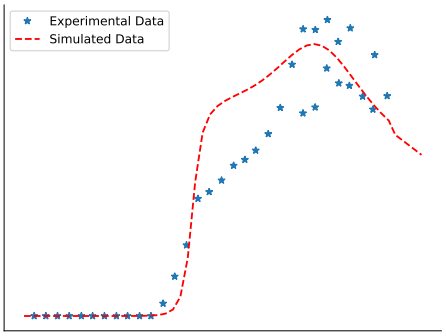

Figure 3: Example of experimental and corresponding simulated data. Simulated data are theoretically a continuous set of points that the predictive model can compute. In practice, simulating a data point could be very expensive.

correctly. In fact, while in this area there is a progressive propensity for sharing and greater availability of data, nevertheless it remains a sector in which the order of magnitude of existing data is much lower than in other areas such as, for instance, that of social media; therefore, it is essential to collect and correctly catalog the experiments in order to encourage their reuse. The domain experts defined which metadata are mandatory and which ones are optional. Thus the relational database schema was designed accordingly. An abstract representation of the experimental data metamodel is provided in Fig. 2. The analysis tools will leverage this metadata to understand the predictive behavior of the model in specific conditions.

In our scenario, the primary source of experiments is journal papers. Inside a paper, usually, there are multiple plots (an example in Figure 3) or data tables about the measurements, where the corresponding metadata are not uniquely tabulated but are described as narrative in the text. Recently, the tendency is to share the numerical data of the experiments in the supplement material associated to papers, facilitating the data collection. In some cases, a representation of the experimental data and metadata is already available in a commonly used format in the domain, such as the XML ReSpeCTh format adopted in [12], and it is available with a DOI associated to it. Metadata for the published papers are extracted from Scopus retrieving citation data using the search APIs[3].

Model Predictive models are treated as black boxes that, if provided to a numerical solver, can predict a particular domain setting. Thanks to the increasing availability of data and computational resources, the number of developed models has increased in the last few years. Nevertheless, not all the models can predict all the environmental conditions of a domain. Therefore, the metadata associated with (but not only) a predictive model is fundamental to study the behavior of a predictive model. The reasons behind the different capabilities of the predictive models vary but are mainly due to computational expensiveness: what is known as a "detailed model" is a complete model and can predict the behavior of a domain in many different conditions, but it takes a long time to execute since it has to solve many differential equations. For this reason, simplified models are derived from the detailed ones with the cost of shrinking the prediction accuracy and reducing the capability to operate and predict all the possible conditions of the domain. As in the case of the experiments, the domain experts define the mandatory metadata for a model (Figure 4).

Simulated Data Simulations connect experiments to models. Given a model and a numerical solver, it is possible to simulate an experiment specifying the experimental condition to the solver, thus generating the corresponding simulated data. These generated data are fundamental to performing different types of analysis on the experiments and on the model. For example, model validation is one of the most critical phases in the model development process. In this procedure, the model performance is evaluated by comparing the similarity of the experimental data with the corresponding simulated data, as in Figure 3, generating one possible type of analysis data.

[3]https://dev.elsevier.com/

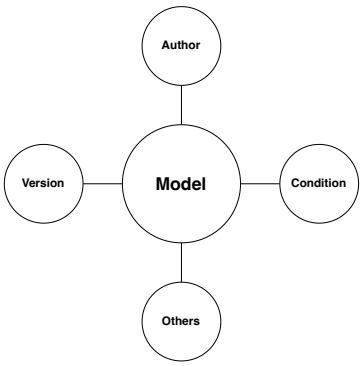

Figure 4: Metamodel of the 'model' data.

3. Framework

Until now, the prototype was a proof of concept of what can be achieved, and once it was delivered, new requirements and discussions arised from the final user. In addition, with the switch to the framework version, new challenges related to day-by-day use needed to be properly addressed.

First, the framework should manage and automate the entire life cycle of the data correctly, from their insertion to the exchange, with all associated implications such as data errors and different representation formats of the data. Second, it is critical to integrate analysis tools to extract knowledge from the data. As before, the design and implementation of the new features have to be done, keeping in mind that the final goal is to create a data ecosystem for experiments and predictive models. A detailed description of the framework is provided in [15]. This section focuses on the most important aspects related to the development of a data ecosystem as a final goal of the project emerged during this phase.

3.1. Data integration and exchange

In some domains, experimental and simulated data could be expensive to generate or replicate. As a result, the data are accumulated over decades (in our domain, some of them are from the late 40's of the past century), witnessing an evolution of the representation formats over the years. Even in the last years, with the digitalization of the data, commonly agreed representation formats can be challenging to develop since it is rare to witness a perfect agreement within the scientific community about what is mandatory to represent.

Interoperability is a fundamental prerequisite for a data ecosystem, and for this reason, the strategy that reconciles the use of many representation formats, thus collecting as much data as possible, is to employ translation engines. Since the possible formats are few in our case study and there is no prevalent representation format. This strategy was the best trade-off. All data inside the framework are only stored in the relational database, following the schema defined by the experts without being bound to use particular formats. In order to feed and collect data from the framework, we need translation engines for every required representation format. Similarly, each numerical solver accepts a configuration file for each simulation and produces an output file in a specific format. Also in this case, the use of translation engines allows to be independent of the representation format of the data.

3.2. Data management

Our data ecosystem has been designed to gather in the same system, models, experiments, and simulations. Thanks to this structure, as shown in Figure 5, the framework acts as a man-in-the-middle that manages and shares the knowledge between the four entities to generate new knowledge.

The downside of this conceptual architecture is that the entities are strongly connected, and incorrect data could quickly impact others. For this reason, inside the framework, it is introduced the concept of ownership of data to contain this hazard. In this way, it is possible, once identified, to quickly identify all the erroneous data involved. Services are provided in the framework for the analysis of data quality and for comparing the results of simulations with experimental data, as described in the following sections. In addition, data management operations on experimental data are provided to improve the quality of the stored experimental basis in the repository. This concept will be particularly helpful in the design of the roles in a data ecosystem as described in Section 4, and therefore regulate access to data.

3.3. Data quality

Nowadays, predictive models are increasingly data-driven, even in domains where a description with physical laws of the phenomena is available. For this reason, data quality plays a more and more central role in the model development process since it directly impacts the prediction quality. In addition, ensuring certain data quality levels within the DE enhances the system trustworthiness, thus starting a loop of increasing the number of users as a consequence of the increased amount of collected data and vice versa.

In our domain, following the concept of fitness for use [16], three quality dimensions have been identified: completeness, consistency, and accuracy. Timeliness is not of interest in the context of experiments and simulations, even if it is often used as a quality metric, mainly for two

reasons: first, even if older experiments are carried out with older and less precise instruments, they still represent a valuable source of information, and their imprecision should be included in their uncertainty evaluation, which it "just" needs to be handled correctly. Second, since the experiments are expensive and hence rare, it is pretty unlikely that multiple experiments are carried out in exactly the same conditions, thus "updating" the old values. For a similar reason, since the predictive models are deterministic, the simulated data does not change over time if forecast with the same model, and numerical configuration of the solver.

In the framework, the data quality control process is composed of two parts, one automatic and the other manual, where the automatic control is performed right after the insertion of a new data in the repository and not, for example, a posteriori based on a recurrent schedule. Data that does not reach the minimum data quality requirements are immediately rejected.

As in all the data quality applications, the rules to measure the data quality dimension depend on the domain, and, often, they are also implementable as automatic checks. Regarding completeness, thanks to the domain knowledge provided by the experts, it is possible to know which metadata is mandatory or optional and in which conditions. For example, it is usually compulsory to express the unit of measurement and the name of the measured property, but for some properties' values, the unit has not been expressed since they are adimensional. Consistency works in a similar way: it is checked that properties of the same instance are consistent with each other. A typical example is an accordance between the property name, like "pressure," and a plausible unit of measurements such as "bar" or "pascal." Finally, the accuracy of the data is considered. It is well known that estimating accuracy is by far the most challenging data quality dimension, but in a framework where experiments and models are combined, it has a non-negligible advantage.

In Figure 5, the typical relation between the experiments and the model is shown: during the model validation procedure, the experiments are used to quantify the predictive model performance. However, since the model is obviously not perfect, it has an (epistemic) uncertainty, but it is reliable enough in many different conditions so that it can be used to check if most of the information inside an experiment is meaningful. In fact, both the accuracy of the numerical data and the metadata could be tested. If the predictions differ significantly from the simulated ones, this discrepancy suggests an error in the reported measurements or in the metadata used to set the simulation. In other words, the model can validate the experiments. This approach foresees cross-validation of an experiment against multiple simulated data about the same experimental condition but using different models.

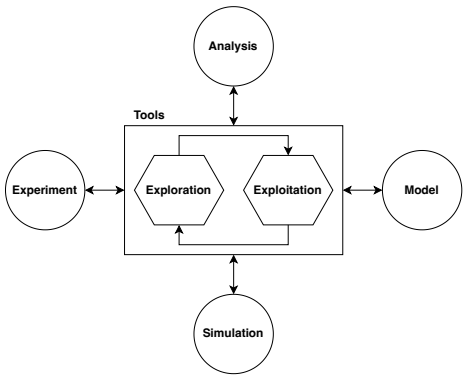

Figure 5: The data ecosystem, with its tools, acts as a man-in-the-middle between the four types of data.

In such a way, it is possible to create a de-facto ground truth against which can assess if an experiment is plausible or wrong. However, this is not always true: if the experimental data are very different from the simulated data, then this is only a hint of a possible error but not a certainty. Therefore this automatic approach is combined with a manual validation of the experimental data by an expert.

3.4. Data FAIRness

FAIR (Findable, Accessible, Interoperable, and Reusable) [17] data have shown to bring many benefits to data ecosystems. In this section, following the recommendation from the literature [18], it is presented appropriate functionalities for each principle of the FAIR policies that have been implemented or designed for the experimental data inside our data ecosystem.

Findable Experiments are stored and used inside the data ecosystem through a relational database that is very flexible and easily maintainable if compared to a file-based organization of the experimental data. Nevertheless, a database representation of the experiments is not findable, and, for this reason, for each experiment, we create an XML representation of the experiment following an XML schema that is widely accepted in the scientific community of experiment's domain. The file is then automatically uploaded to Zenodo to assign to it a unique global identifier together with other metadata that make the experiment searchable without necessarily using our data ecosystem.

Accessible Experiments inside our data ecosystem are identified both with a (numerical) primary key and the associated DOI. A primary numerical key makes imple-

menting the relational instances in the database easier even before the DOI has been generated. Our data ecosystem offers data management services through a HTTP API, accepting typical formats of the request such as CSV, JSON, and XML. One of the advantages of such HTTP API micro-services structures is that the final users are not requested to use a particular software or programming language or technical expertise to access data and services, and they can combine them as preferred. Authentication is required to use the API upon a free sign-in request procedure. Authentication enables traceability and accountability of the operations and helps keep a quality level of the scientific repository with respect to an open-access configuration.

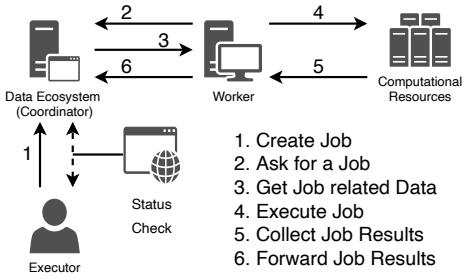

Figure 6: Coordinator-worker architecture.

Interoperable Experiments in their XML representation format are a plug-and-play solution. Every researcher can use them as preferred, paying attention to the definition of each XML tag. If the experiments are accessed through the HTTP API, the same vocabulary of the XML representation format is used to query the database and for the responses.

Reusable One of the primary purposes of the data ecosystem is to reuse data, encourage their sharing among institutions and avoid duplicates. Experimental data can be uniquely cataloged through some metadata. Developing the database around the uniqueness constraint of these metadata allows us to maximize the reuse.

3.5. Data generation and analysis

Thanks to the model, we can theoretically generate an infinite number of simulated data, and similarly, using the analysis tools and combining them as we prefer, we can create a vast number of analysis data. Neglecting the space needed to store such quantities of data, the first limitation that makes this idea unfeasible is the amount of computational resources needed to generate them. A centralized architecture where all the computational burden is on a single organization is not sustainable. Even if the cost is shared, the bureaucracy behind sharing computational resources is very complicated. The solution to this problem is a coordinator-worker architecture where the framework, i.e., the coordinator, collects the jobs and distributes them among the workers, that in some cases can delegate the job to other machines as shown in Figure 6. The coordinator-worker configuration is scalable and allows each user to decide how many computational resources to dedicate and use only for their jobs.

Providing analysis tools inside the framework is a game-changer. The user is incentivized to stay in the system and leverage the other knowledge in terms of data and tools available. The more the users stay in the system the more they are inclined to share data, thus improving the overall system and starting a virtuous circle. Such tools generate new knowledge about the data or the domain and increase the awareness and insight on data. In fact, it is central, for example, the concept of database coverage or diversity. In all data-driven models, "you are what you eat," and therefore if a model is generated only using data that represent a restricted portion of a domain, the model will be able to more or less correctly predict only what it has already seen. Drawbacks of such an approach could lead to ethical problems since classification, and regression models could have strong biases based on the diversity and the balance of the data used to generate them. A predictive model for physical domains suffers from the same hazard: data are mainly used for the validation phase. If the model is validated against a large amount of data but not diverse, the predictive model performances could be astonishing, but in practice, they could be much worst.

4. Data ecosystem

The final stage of this evolution regards the transition from a framework to a data ecosystem. In this last evolution stage, what is important to investigate is how the framework that has been actively used by one research group should evolve to host multiple organizations and many more users. This transition that seems straightforward in practice has mainly two different challenges that can be smoothly implemented thanks to the designed choices of the previous project steps. First, activities for the repository management described before, such as experiment validation, need to be formalized in terms of responsibility and accountability. Second, the data ecosystem could host data with intellectual properties (IP) that are not yet open access but are on the data ecosystem because the final user wants to take advantage of our functionalities and analysis tools to compare, for example, the quality of data. Both these challenges have in

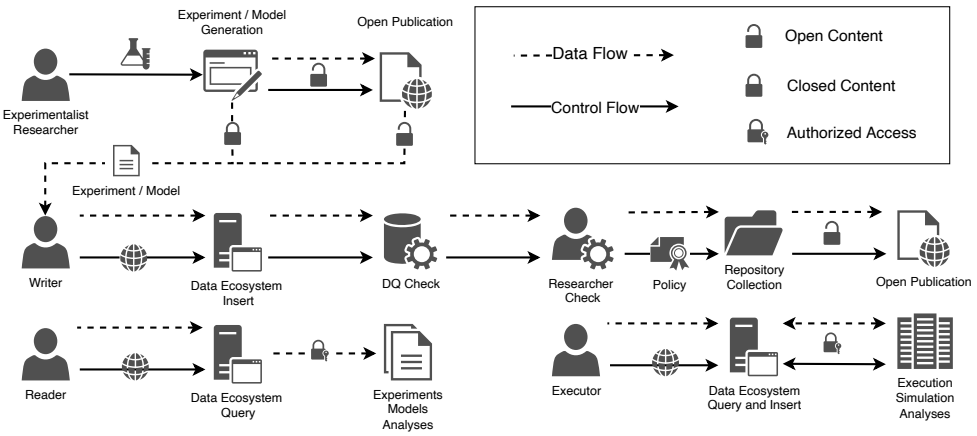

Figure 7: The main five roles inside our DE, and their main four activities.

common that it is necessary to define user roles and rules with corresponding permissions over the data ecosystem functionalities.

In this scenario, it is assumed that the data ecosystem is trustworthy in terms of privacy and security, and any specific entity does not own it, but it belongs to the community.

4.1. Roles

Several organizations collaborate within a data ecosystem. An *organization* is an abstract concept that groups several people. Sometimes it is possible to map this concept to other familiar entities such as a university, a research center, a department, or a research group. Each user belongs to at least one organization to be part of our data ecosystem and has at least one role. The (virtual) ownership of the data belongs to the organizations. Data entered or generated by a user will be owned by the organization to which it belongs, while the paternity of the data remains to him/her. The users must specify whether the data deriving from them are open or closed content. Each user can access all the open-content data of all organizations inside the data ecosystem, and all the closed-content data belonging to their organization(s).

The configuration in organizations allows an easy share of closed-content resources among them with different levels of granularity and relationship: a single experiment or a group of them could be shared with another organization, or an organization can share in one direction or both directions the whole closed-content data.

The data ecosystem holds the role of *publisher*: as soon as a content item is made open, the DE generates, in the case of experiments, an XML representation file that is published in Zenodo[4] to associate a DOI to it and enhance accessibility and findability.

Besides the publisher role, in our scenario, it was identified five user roles as follows. Figure 7 shows the five roles involved in four typical actions in the overall workflow for the model development process. The actions represented are the experiments or models generation and insertion into the DE; the collection of data, such as analyses, experiments, simulations, and models, together with the creation of simulation and analysis jobs.

Experimentalist This role identifies a scientist that carries out the experiment and generates the experimental data. The experimentalist has the intellectual property of data. Based on the situation, the experimentalist can decide to immediately publish the results in a journal (or similar) or provide the data directly to other entities through a private communication and publish them later. Accordingly to this choice, the experiments have an open or closed content policy, respectively. Even if a journal is not open access or requires a subscription, its experiments are considered open content because they are publicly available material.

Researcher The researcher has mainly two functionalities in our DE and scenario. First, it generates the predictive model, and, as in the case of the experimentalist, it has the faculty to choose the publication policy. Second, it has the duty to verify the experiments in their validation procedure as described before. Suppose the experiment that has to be validated is open-content. In that case, a cross-validation strategy is preferred: a researcher from a different organization of the experiment own-

[4] https://zenodo.org/

ership will perform the task to avoid possible bias and enhance the DE's overall trustworthiness. It is assumed that there is at least one researcher per organization.

Reader The reader represents the user that has permission to access the open contents and all the closed contents belonging to its organization. Thanks to the authentication, transparently, it is possible to hide part of experiments, models, simulations, and analyses without changing the API.

Writer The writer is a trained user that has the task to insert into the DE all the collected data. It is a trained user because, for this field, it is not a straightforward operation and it requires basic domain knowledge, even if the system and the researcher will check their validity later. The writers mainly insert experiments and models. They can find these data in the literature, or they can be provided through private communication. In any case, it is their responsibility to associate the correct content policy to objects.

Executor This role represents a kind of user that has the privilege to allocate resources and generate new data in terms of simulations and analyses. In both cases, the executor needs to have access to both experiments and models to create a new simulation or perform analyses (like in the case when it is needed to compare experiments against simulations). This kind of operation could result in expensive operations. Also, in this case, domain experience is required, for example, to set the optimal numerical configuration to solve a simulation numerically and thus use the computational and storage resources wisely.

It is worth mentioning that even if an experiment is closed-content and the user has not the permissions, its metadata, i.e., in this domain, the experimental condition, is in any case open, and therefore it is possible to simulate this configuration. Nevertheless, all the analysis operations concerning comparing the simulated data against the experimental data will be hidden.

5. Discussion and Conclusion

In this paper, it was presented our experience in developing a data ecosystem to improve the development process of a chemical-physical predictive model. As happened often in practice, our design process of the data ecosystem was a bottom-up approach rather than a top-down due to the necessity of delivering a usable product quickly. The development of the final system foresees three product-related phases: prototype, framework, and data ecosystem. In each step, some properties of the final data ecosystem are taken care of. This approach allowed us to increasingly add complexity to the final ecosystem's design and deal with new requirements arising from a non-typical application domain more smoothly. In addition to the typical challenges, a chemical engineering data ecosystem has to deal with a specific type of data, such as predictive models and experimental and simulated data, that require ad-hoc methodologies, for example, in the case of data quality measurements or intellectual property management. Some of these aspects are distinctive of scientific repositories, while the three-phase approach and some challenges and solutions are more universal. The prototype phase, in particular, is important to collect the requirements arising from a new and complex domain with the final goal of discovering the main types of data that need to be stored and the necessary services. The result of this step is the database and system architecture. A micro-service structure is a convenient architecture since, during a bottom-up approach, it is very probable that new requirements will arise. Implementing a new service will be a combination of the existing ones. The framework step addresses the challenges of transforming a proof-of-concept into a system used daily by a restricted number of users. Therefore, this system version accounts for data quality and management aspects, implements FAIR principles, and has to be scalable in terms of computational resources. The final evolution deals with distinguishing the user roles inside the DE and data ownership. In such a way, it is guaranteed higher trustworthiness and transparency of the system and of the data while fulfilling the intellectual property requests. Future developments concern the improvement of the implementation of some FAIR principles, in particular findability and reusability. We plan to introduce new features to allow from outside to make searchable experiments with a restricted access policy due to their intellectual property. Exposing just the metadata could enhance both the findability and reusability of the experiments. In addition, we plan to present a provenance data model to improve the reusability of the analyses and the models, following the W3C recommendations.

References

[1] M. Jarke, B. Otto, S. Ram, Data sovereignty and data space ecosystems, Business & Information Systems Engineering 61 (2019) 549–550.

[2] E. Curry, A. Sheth, Next-generation smart environments: From system of systems to data ecosystems, IEEE Intelligent Systems 33 (2018) 69–76.

[3] V. Stodden, The data science life cycle: a disciplined approach to advancing data science as a science, Communications of the ACM 63 (2020) 58–66.

[4] C. Tenopir, E. D. Dalton, S. Allard, M. Frame, I. Pje-

sivac, B. Birch, D. Pollock, K. Dorsett, Changes in data sharing and data reuse practices and perceptions among scientists worldwide, PloS one 10 (2015) e0134826.

[5] C. Cappiello, A. Gal, M. Jarke, J. Rehof, Data Ecosystems: Sovereign Data Exchange among Organizations (Dagstuhl Seminar 19391), Dagstuhl Reports 9 (2020) 66–134. URL: https://drops.dagstuhl.de/opus/volltexte/2020/11845. doi:10.4230/DagRep.9.9.66.

[6] F. Sidi, P. H. S. Panahy, L. S. Affendey, M. A. Jabar, H. Ibrahim, A. Mustapha, Data quality: A survey of data quality dimensions, in: 2012 International Conference on Information Retrieval & Knowledge Management, IEEE, 2012.

[7] E. Ramalli, B. Pernici, Know your experiments: interpreting categories of experimental data and their coverage, in: SeaData at VLDB 2021, CEUR Workshop Proceedings, 2021, pp. 27–33.

[8] I. N. Chengalur-Smith, D. P. Ballou, H. L. Pazer, The impact of data quality information on decision making: an exploratory analysis, IEEE Transactions on Knowledge and Data Engineering 11 (1999) 853–864.

[9] M. Drosou, H. Jagadish, E. Pitoura, J. Stoyanovich, Diversity in big data: A review, Big data 5 (2017) 73–84.

[10] S. Geisler, M.-E. Vidal, C. Cappiello, B. F. Lóscio, A. Gal, M. Jarke, M. Lenzerini, P. Missier, B. Otto, E. Paja, B. Pernici, J. Rehof, Knowledge-driven data ecosystems: Towards data transparency, ACM Journal of Data and Information Quality 14 (2022) 1–12.

[11] M. Frenklach, Process informatics for combustion chemistry, in: 31-th International Symposium on Combustion, Heidelberg, 2006.

[12] T. Varga, T. Turányi, E. Czinki, T. Furtenbacher, A. Császár, Respecth: a joint reaction kinetics, spectroscopy, and thermochemistry information system, in: Proceedings of the 7th European Combustion Meeting, volume 30, Citeseer, 2015, pp. 1–5.

[13] C. Batini, M. Scannapieco, Data and information quality: Dimensions, Principles and Techniques, Springer, 2016.

[14] G. Scalia, M. Pelucchi, A. Stagni, A. Cuoci, T. Faravelli, B. Pernici, Towards a scientific data framework to support scientific model development, Data Science 2 (2019) 245–273.

[15] E. Ramalli, G. Scalia, B. Pernici, A. Stagni, A. Cuoci, T. Faravelli, Data ecosystems for scientific experiments: managing combustion experiments and simulation analyses in chemical engineering, Frontiers in Big Data (2021) 67.

[16] R. Y. Wang, D. M. Strong, Beyond accuracy: What data quality means to data consumers, Journal of Management Information Systems 12 (1996) 5–33.

[17] M. D. Wilkinson, M. Dumontier, I. J. Aalbersberg, G. Appleton, M. Axton, A. Baak, N. Blomberg, J.-W. Boiten, L. B. da Silva Santos, P. E. Bourne, et al., The fair guiding principles for scientific data management and stewardship, Scientific data 3 (2016) 1–9.

[18] H. Koers, D. Bangert, E. Hermans, R. van Horik, M. de Jong, M. Mokrane, Recommendations for services in a fair data ecosystem, Patterns 1 (2020) 100058.