Data Science 1 (2016) 1–5
IOS Press

Scholarly data analysis to aid scientific model development

New requirements emerging from the chemical kinetics domain

Gabriele Scalia [a,*], Matteo Pelucchi [b], Alessandro Stagni [b], Alberto Cuoci [b], Tiziano Faravelli [b] and Barbara Pernici [a]

[a] *Department of Electronics, Information and Bioengineering, Politecnico di Milano, Piazza Leonardo da Vinci 32, 20131 Milano, Italy*
E-mails: gabriele.scalia@polimi.it, barbara.pernici@polimi.it
[b] *Department of Chemistry, Materials, and Chemical Engineering Giulio Natta, Politecnico di Milano, Piazza Leonardo da Vinci 32, 20131 Milano, Italy*
E-mails: matteo.pelucchi@polimi.it, alessandro.stagni@polimi.it, alberto.cuoci@polimi.it, tiziano.faravelli@polimi.it

Abstract. The sharing of scientific and scholarly data has been increasingly promoted over the last decade, leading to open repositories in many different scientific domains. However, data sharing and open data are not final goals by themselves, while the real benefit is in data reuse, which allows leveraging investments in research and enables large-scale data-driven research progresses. Focusing on reuse, this paper discusses the design of an integrated framework to automatically take advantage of large amounts of scholarly scientific data to support research, and in particular scientific model development. Scientific models reproduce and predict complex phenomena and their development is a rather challenging task, within which scientific experiments have a key role in their continuous validation. Starting from the chemical kinetics domain, this paper discusses a set of use cases and a first prototype for such a framework which lead to a set of functional requirements and an architecture that can easily be generalized to other domains. The paper analyzes the needs, the challenges and the research directions for such a framework, in particular those related to data management, automatic scientific model validation, data aggregation and data analysis, to leverage large amounts of scholarly data for new knowledge extraction.

Keywords: scientific model development, experimental data, scholarly data, scientific model validation, chemical kinetics

1. Introduction

Scientific repositories allow the collection and distribution of scientific and scholarly data. Among the fundamental factors to their growth there is an increasing availability of computing power and storage capacity and of new *big data* analysis techniques to manage such vast amounts of data. However, since their initial spreading, some challenges have been highlighted in the literature, like availability, integration, extensibility and maintainability [32].

The urgent need of improving the infrastructure supporting the *reuse* of experimental data has been highlighted in the literature and led to general guidelines to create and manage repositories, notably the FAIRness [54] (being findable, accessible, interoperable and reusable) or the "pyramid" of needs for data

*Corresponding author. E-mail: gabriele.scalia@polimi.it.

management that span from being simply *saved* to being *shared* to ultimately being *trusted* [18]. The collection of scholarly data is becoming more and more important for the validation of research results. One of the current goals is to improve the capability of sharing experimental data among researchers, in order to enhance their quality and reproducibility and to derive new research results. Based on the Open Science Cloud strategy of the EU[1], *every* project financed within the H2020 framework has to deal with the FAIR data policy.

The sharing of scientific data has been increasingly promoted over the last decade. However, it has been recently discussed how "data sharing practices, especially motivations and incentives, have received far more study than has data reuse" and that data sharing and open data are not final goals by themselves, while the real benefit is in data reuse, which is "an understudied problem that requires much more attention if scientific investments are to be leveraged effectively" [35]. Reusing allows leveraging investments in research and enables large-scale data-driven research progresses. However, in order to reuse datasets from multiple sources, challenges like integration must be addressed.

This paper focuses on reuse, and in particular on the design of an integrated system to automatically take advantage of large amounts of scholarly scientific data with the goal to support research. This requires the integration of data management and analysis techniques with the scientific development workflow. If a scientific repository with its tools can improve the efficiency of many tasks, like searching and exporting selected entries, the usage of new data analysis techniques can leverage the large amounts of integrated scholarly data to automatically and dynamically extract new hints which are normally not obtainable manually or through partial datasets given the large volume of data required.

One key research activity in many scientific domains is *scientific model development*, where the goal is to build models to reproduce and predict complex phenomena. Scholarly data are largely used within this activity and, in particular, experimental data have a crucial role. Indeed, extensive comparison of models simulations with real experimental data is needed in the iterative scientific model development life cycle, to continuously validate and improve the underlying models.

In this paper, we focus on experimental data and model development in the combustion domain, and in particular in the *chemical kinetics* domain, where a number of initiatives to collect experimental data in a systematic way have been already developed. Examples of data repositories in this domain are ReSpecTh [50], CloudFlame [26], ChemKED [53] and PrIMe [22]. They aim to collect and store the increasing number of basic and complex experimental measurements of combustion phenomena (or properties) available in the literature in more efficient machine-readable formats (e.g., XML, YAML, and so on). In parallel, EU-funded projects are pursuing the challenging goal of defining community data reporting standards in this domain[2] to overcome instances of incomplete, inaccurate, or ambiguous descriptions of fundamental data, both in past and in recent scientific literature. Moreover, key steps in the automatic development and validation of combustion kinetic models are strongly emerging in recent papers [7, 24, 29, 49]. Even if this paper focuses on a specific domain, the use cases, the requirements and the solutions analyzed can be shared among many different domains, both in other scientific fields and in other different areas, as discussed in the following.

Automatically taking advantage of scholarly data to aid model development brings several challenges:

- First of all, data not only need to be collected, but also *semantically interpreted* and *integrated* through domain-specific ontologies. Taking into consideration the semantics of the stored data requires "machines to be capable of autonomously and appropriately acting when faced with the

[1] https://ec.europa.eu/research/openscience/index.cfm?pg=open-science-cloud
[2] http://www.smartcats.eu/wg4/task-force/

wide range of types, formats, and access-mechanisms/protocols that will be encountered during their self-guided exploration" [54].
- *Data quality* must be ensured, since it has an impact on the whole model development cycle and a low data quality could bring to wrong results. This includes also *data cleaning* activities.
- Objective and *quantitative measures* to assess the performance of a model with respect to some experimental data are necessary [6].
- Several dynamically evolving sources of data must be taken into account for complex analysis tasks.

The goal of the present paper is to discuss the design of a system which leverages scholarly data to aid scientific model development, describing a set of use cases with the related challenges and presenting new requirements emerging from a working prototype and an integrated architecture to support them. The domain considered in this paper is kinetic modeling of chemical processes such as combustion [41, 50], but requirements and solutions are general and may be extended to many other fields in the wider domain of scientific modeling.

The novel contributions of the present paper are as follows:

- New requirements for an integrated system which leverage large amounts of scholarly data to support scientific model development are discussed, starting from a set of use cases in the chemical kinetics domain and a working prototype.
- A data-based and service-based architecture for such a system is proposed and discussed, providing a data model to support data integration and the development of a set of data curation and analysis services.

The present paper is an extension of [44], presented at the SAVE-SD workshop on *Semantics, Analytics, Visualisation: Enhancing Scholarly Dissemination*, co-located with The Web Conference 2018. While [44] focuses on new requirements and research directions arising from the limitations of a first prototype, the present work discusses the design of a new integrated framework to support scientific model development starting from the revised requirements emerging from an extended working prototype and the analysis of new use cases.

The paper is structured as follows. Section 2 introduces the domain presenting the scenario, Section 3 discusses related work, Section 4 describes a set of use cases for the system and the developed extended prototype, Section 5 discusses the emerging requirements and Section 6 presents an integrated architecture to support them. Finally, Section 7 discusses perspectives for future research directions.

2. Scenario

In this section we describe the scenario: kinetic modelling of combustion. Since the requirements which arise from this scenario are shared among many other domains, the section ends with a discussion on its generalization.

Combustion kinetic modelling has been driving the development of more efficient fuels and combustion technologies for the last 40 years. Chemical kinetics determines the reactivity of a given fuel or a fuel mixture; thus, a better understanding of the effects of a specific chemical compound on combustion performances and emissions allows the tailoring of a fuel or a fuel blend for an existing infrastructure or *vice-versa* [5].

The development and update of reliable kinetic models is a rather challenging task, directly reflecting the intrinsic complexity of combustion phenomena, and it is one of the fields of research of the CRECK

modeling group[3] [40]. Such models typically involve $\sim 10^2 - 10^3$ chemical species connected by a network of $\sim 10^3 - 10^4$ elementary reaction steps. Moreover, a combustion kinetic model hierarchically develops from small chemical species (e.g., hydrogen, methane, and so on) up to heavier compounds typically found in commercial fuels such as gasoline, diesel, jet and alternative fuels. For this reason, any modification in the core computational model significantly propagates its effects to heavier species making continuous revisions and updates mandatory to preserve the reliability of the model.

The OpenSMOKE++ software [17] has been developed to execute such models performing kinetic simulations of typical facilities such as jet stirred and flow reactors, 1D-2D laminar flames, shock tubes and rapid compression machines, specifically conceived to decouple purely chemical kinetics effect from fluid dynamics, heat and mass transfer phenomena. The variables of interest are typically ignition delay times, laminar flame speeds, fuel/oxidizer mixtures, fuel consumption, intermediate and product species formation/disappearance at specific conditions of temperature (T), pressure (p) and dilution.

From an operational perspective, the iterative validation of such models (Figure 1) strongly relies on extensive comparisons of results from model simulations with experimental data covering conditions of interest for real combustion devices. The key step in such procedure is the *assessment* of model performances with respect to experimental data. Experimental data are typically stored in *supplementary materials* attached to scientific papers as CSV, ASCII or excel files. While this good reporting practice is almost always verified in recent publications, older studies were not used to provide such a detail. In this case, semi-automated or automated extraction of experimental data and boundary conditions from tables, figures and text reported in the manuscript is necessary. The assessment is typically performed using plots in which the experimental data and the curves from model simulations are plotted together and the researcher express his/her subjective evaluation based on its experience and knowledge, often neglecting experimental and model uncertainties. This standard model validation procedure has been recently defined as a "poorly posed" problem [51]. Analysis tools (e.g., sensitivity analysis) allow highlighting relevant model parameters and drive their refinement by means of more accurate estimation methods.

From a data perspective, an experiment consists of a set of *conditions*, which describe the experimental setting, and a set of *output variables*. The conditions can be categorical values (e.g., the reactor type), constants (e.g., the initial mixture or the initial pressure of the experiment) or, less frequently, variables. The conditions are the only needed to replicate an experiment or simulate it with a model. Output variables are given as one variable functions, and therefore can be represented as curves; an example is the evolution of chemical species over time.

In order to automate the performance assessment of a model and to derive objective and quantitative measures, overcoming traditional graphical comparisons, the "Curve Matching" algorithm [6] has been recently developed. It allows an objective, quantitative and automatic evaluation of the model capability to predict the variables of interest, extending the most common, but sometimes misleading, sum of squared error based approach [33].

If the model provides satisfactory agreement, subsequent steps of optimization and reduction [48] make the model suitable for large scale computations of interest for industry industry (i.e., reactive computational fluid dynamics). On the contrary, if the model shows deviations outside of the experimental uncertainties, relevant pathways that can be identified by means of analysis tools and model parameters are further refined with better estimates. Indeed, the recent developments coupling high

[3] http://creckmodeling.chem.polimi.it/

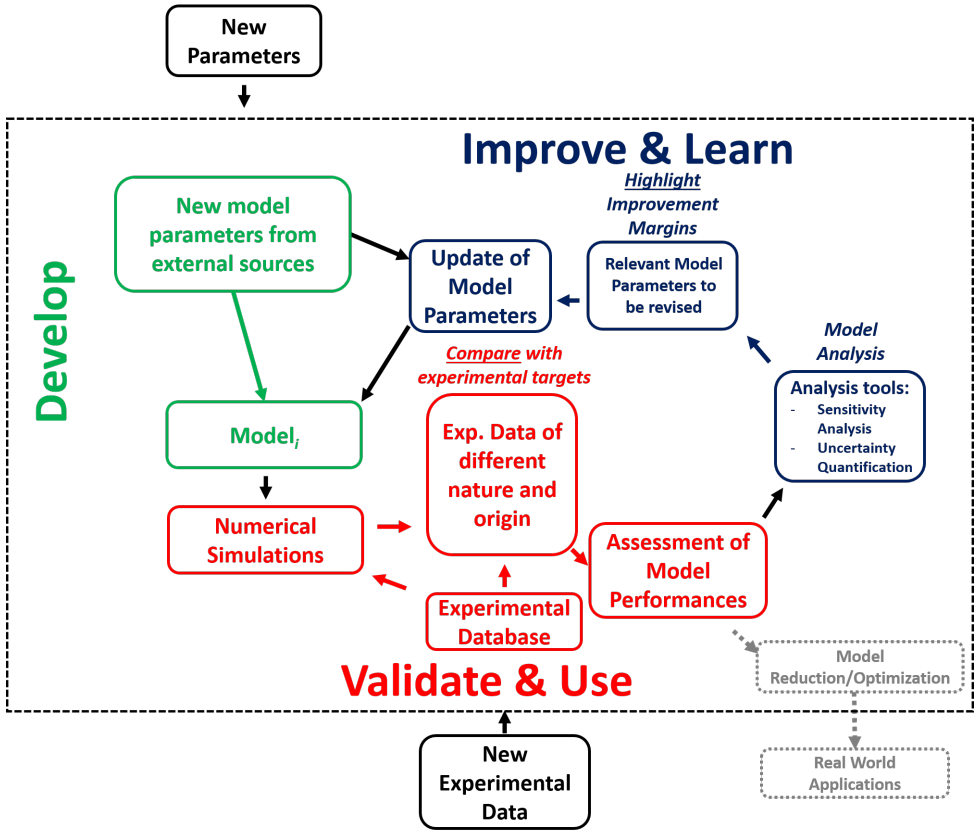

Fig. 1. Standard development, validation and refinement procedure of a chemical kinetic model for combustion applications [38].

performance computing and theoretical chemistry allow the automatic generation of highly accurate parameters [7, 11, 24, 29, 49].

While the efficient integration of the above tools in a fully automatized system is one of today's challenges in kinetic modelling [29], efficient and smart data collection, formatting, analysis, conversion and storage constitutes step "zero" within the domain. The exponential growth in the volume and the complexity of scientific information in the combustion community (experimental data, models, theoretical investigations, and so on) and the improved accuracy of experimental techniques can be beneficial at best only if coupled with extremely efficient tools for acquiring, storing and analyzing such information, therefore allowing real advances in knowledge.

Several initiatives to enable effective and structured data collection of experimental data for combustion science are available in the literature at present. Starting from the pioneering work of M. Frenklach and co-workers developing the PrIMe database [22], which is still under continuous update, the ReSpecTh repository [50] largely improved and extended it by means of a more flexible, detailed and user-friendly data structure. At present ReSpecTh has collected more than 1,000 experiments into XML files, which correspond to more than 87,000 data points, and the extent of this collection is expected to increase in the years to come. CloudFlame (KAUST) [26] provides an open database for experimental data in a standard CSV format, together with a cloud infrastructure for running simulations based on stored models and data. To improve human readability of structured data, ChemKED [53] has intro-

duced a human and machine readable data standard for chemical kinetics experiments based on YAML, together with a set of validation and conversion tools. The COST Action CM 1404 (SMARTCATs[4]) established a task force of scientists aiming to define standards for data collection, allowing easy and effective coupling with the above systems.

On top of the reference repositories mentioned above, one should consider a large amount of experimental information stored in less structured formats into many institutional servers belonging to experimental or modelling groups working in the field of combustion. As an example, the CRECK repository is the result of data collection in ~ 40 years of research efforts in modelling combustion kinetic and thermal processes [40]. While data management has relied on manual extraction and classification into unstructured spreadsheets or text files for a long time, recently a new relational database with an interface for querying and exporting data has been developed [41]. In its "beta" version it contained references to 30 scientific papers corresponding to 1,000 data points and guaranteed interoperability with the ReSpecTh data standard. Limitations of this first prototype and new requirements [44] led to the design and realization of a new system which includes not only an experimental database, but also other integrated tools to support the whole model development cycle and is the subject of this paper.

The requirements arising from the scenario described up to now can be easily generalized to other domains and contexts. The need of a continuous validation of models based on new experiments is shared among most scientific fields, and therefore the activities of acquiring, analyzing and evaluating models and experiments is certainly shared with other scientific domains.

3. Related work

In the following, some works related to the present work are described. Given the fact that the proposed framework includes challenges related to several topics, including data management, quality and cleaning, each of them is briefly outlined, before describing other systems with similar goals and other related works which motivated this system. Talking about requirements on specific issues, other papers will be cited in the following sections.

All the requirements analyzed in this paper strongly rely on *data management* efforts. Indeed, it has been discussed in the literature how value creation is not driven by data itself, but by the whole data management process [58], and that it is enabled by the management of both internal and external data. Data reuse, in particular, requires specific management techniques.

The goals of this work stem from the increasingly availability of open and structured scientific/scholarly data and the resulting needs of new ways to effectively leverage and reuse them. In general, the concept of "open data" is not easily definable and baseline conditions are "fewest restrictions" and "lowest possible costs" [36]. It has been noted how data sharing practices have been investigated more than data reuse [35]. Data reuse is not be limited to reproduce research, which is an example of indipendent reuse (that is, reuse of an individual dataset) but includes also *data integration* "to make comparisons, build new models or explore new questions altogether" [35]. This is the direction of the present work, which focuses on reuse as a mean to extract new knowledge not available analyzing individual instances. To this goal, it has been highlighted how standards and formats for data release influence reuse opportunities. Standards are a prerequisite to understand and integrate datasets, but even with them in place, differences in parameters, instrumentation and other factors make data integration not trivial [8]. Moreover, semi-structured data formats could add ambiguities to standardized datasets.

[4] http://www.smartcats.eu/

Challenges related to data reuse and value creation are discussed in [28]. Finding useful datasets is a challenge by itself, which precedes their interpretation. This require adequate metadata and often some kinds of standards, but "adequate" has a different meaning when the goal changes, and often a publication includes only the bare minimum required for its specific goal. There are also format issues, which sometimes can be simply a question of performing a schema mapping, but often require to resolve more substantial mismatches to have an effective integration.

Two fundamental aspects of the proposed framework are data quality management and data cleaning, which are strongly related to the *veracity* of big data. Their importance come from the reuse of data which can be noisy, outdated, misleading and, in general, unreliable. Indeed, incorrect information is normally found in shared data [8] and unreliable experimental data have been identified even recently in the chemical kinetics domain [27]. Anyway, quality issues could be introduced even later in the processing pipeline, for example when experimental data are imported in a repository or converted in a specific format, given also the constraints of the related standard. Independently from the cause, it is important to assess data quality before using data in order to get meaningful results.

Data quality management techniques used in traditional databases are not enough to handle a big data scenario with heterogeneous sources, which instead require an "adaptive approach able to trigger the suitable quality assessment methods on the basis of the data type and context in which data have to be used" [3]. Among other challenges, this requires *context-dependent* quality assessment, which takes into account, for example, that a large number of sources makes trust, and *multi-granularity* assessment, to evaluate data quality at various aggregation levels [10]. Data cleaning techniques allow detecting and repairing data errors by identifying, for example, integrity constraints, duplicates, functional dependencies, and so on. These activities can be automatic or human guided, and error detection techniques can be applied on the original database or later on in the data processing pipeline [13]. In the context of the present paper, the discovery of errors after some processing and integration is of particular importance given the use of large-scale cross comparisons and the integration of external sources. Such "delayed" cleaning has been analyzed and formally described in the literature, for example in [12]. Data integration, and in particular the analytical querying of an integrated set of data sources, is normally addressed through DW and OLAP systems. However, in a traditional OLAP system all the data sources should be known and described in advance, as the rules used to perform their integration, and a traditional ETL process does not allow a continuous integration, especially when there are external sources. Moreover, traditional OLAP systems typically do not deal with semi-structured data and external ontologies. An *exploratory OLAP* supported by semantic technologies can overcome these limitations allowing the exploration of "new data sources, of new ways of structuring data, of new ways of putting data together, of new ways of querying data" [2]. Such solutions provide several benefits with respect to data extensibility and dynamicity, like the capability of handling semi-structured sources and freshness of the results. Their "open-world" assumption allows analytical querying of new external data.

The framework investigated in this paper shares some features with scientific workflow management systems. Workflows are "a set of interrelated computational and data-handling tasks designed to achieve a specific goal" [31] and a workflow management system (WMS) aids in the automation of those operations managing their execution and information exchange. Scientific workflows are typically *data-intensive* workflows and scientific WMSs have been proven crucial in data-driven science [4]. In this paper, we focus on the use cases, the requirements and the challenges coming from the reuse and the large-scale analysis of scientific and scholarly data to extract new insights, in particular with respect to scientific model development. We do not directly address the development of new models or new experiments, under the assumption that these activities occur externally to the framework. The discovery of

new knowledge by systematically processing large collections of experiments and simulations has been addressed also by scientific WMSs [31], which, however, are mainly focused in the global workflow and in related problems such as scheduling and resource allocation in distributed environments. Indeed, a workflow has several computational tasks linked by data dependencies and their scheduling involve challenges like resource provisioning, performance variation, parallelism and fault tolerance [19, 43]. Therefore, our work is complementary to a scientific WMS, since the analysis tasks identified could be managed efficiently in a large-scale and multi-user scenario as scientific workflows.

The management of big scholarly data, which are a key source for the framework, has been surveyed in [55] and currently there are initiatives to publish open bibliographic citation information as linked data, such as Open Citations [39]. The feasibility of tools to explore such data has been demonstrated, for example in [21, 34], facing challenges related to statistical analysis, semantic exploration and visual analytics. Scholarly data analysis is often based on knowledge graph analysis.

The usage of graph structures as conceptual-modeling allows storing and querying data based on the relationships among objects, which is necessary to model inter-dependencies among entities and for data exploration, also through faceted search [42]. Graphs can also be exploited for mining activities [42] and to model varying precision and accuracy. Graph modeling can be accomplished through a graph database [30] or via mappings that enable graph reasoning over traditional relational databases [9, 45].

The large-scale validation of models with respect to experimental data is an emerging topic and other integrated systems have been proposed very recently in different domains. PRISMS [1] is an integrated framework for accelerating predictive structural materials science. It includes computational modules to predict microstructural evolution and mechanical behavior of structural metals and a set of integrated scientific "use cases" where they are linked to experiments. It also includes a collaborative repository to archive and disseminate experiments and computational models. CaRMeN [25] is a tool that automates the workflow of comparing chemical kinetics model and experiments, overcoming the issues of a manual, time consuming and error-prone validation. It contains reactor solvers for different kind of reactors, different models to simulate the catalytic partial oxidation of methane and the related experimental data for validation. Model performances assessment is based on parity plot diagrams and the graphical interface makes it very much user friendly. The present work shares several goals with these projects, but our efforts are directed more towards the management of challenges coming from the use of heterogeneous and potentially noisy data sources and their semantic interpretation to enable new analysis directions, obtaining requirements which are not domain specific.

Recent research results on the generation of knowledge from the large-scale analysis of experimental data in the combustion kinetic domain have been recently reported in [27]. In this work, a collection of 55 experimental datasets in premixed laminar flames extracted from previous publications has been collected showing how their analysis allows the extraction of fuel-specific chemistry that can be useful to identify inconsistent data. Along the same lines, [37] also collected and reviewed a large set of experimental data (60 datasets) on rich laminar premixed flames of hydrocarbon fuels reported in recent years, and analyzed them by means of the available kinetic models to describe the formation of polycyclic aromatic hydrocarbons. The Curve Matching approach [6] was applied in this case to evaluate model performances. This work allowed identifying inconsistencies not only between single models and experiments, but also between different models often describing the same phenomena in significantly different ways, still obtaining a comparable degree of agreement. From these recent examples it is clear how a fully integrated framework with large-scale capabilities, as that described in this work, could be beneficial to both model development and experimental evaluation.

4. Towards an integrated framework

Model development and automatic validation through scholarly data has several facets. In order to analyze the directions of an integrated system with this goal, the main use cases have been analyzed in Subsection 4.1. Moreover, a first extended prototype has been developed and is shown briefly in Subsection 4.2, and will be the base for the formulation of new requirements and an architecture for the complete framework.

4.1. Use cases

4.1.1. Experiment simulation

An experiment that is stored in the framework can be automatically simulated by a stored model. Simulating an experiment means executing the model at the experimental boundary conditions (initial temperature, pressure, fuel/oxidizer mixture composition, residence time, reactor, reactor type, and so on), so that the simulation environment is as much as possible similar to the actual experiment, and comparing the results.

The simulation requires a *simulation software*, used as an external service, such as OpenSMOKE++ [17]. The simulation software takes in input the target model and the experimental boundary conditions and returns the output variables as predicted by the model.

Therefore, automatically testing a model on an experiment requires:

(1) to build a *simulation input* for the simulator, which specifies i) experimental boundary conditions and ii) the target model;
(2) to execute the simulator with the simulation input;
(3) to compare the model and the experiment outputs.

Steps 1 and 3 involve several sub-steps like validation, transformations and post-processing.

The final comparison results in a *similarity score* for each experiment/model outputs which expresses how much they match, that is, the ability of the model to correctly approximate the experimental results. The design of such matching techniques is out of the scope of this paper. Currently, the framework uses the Curve Matching algorithm [6] for this scope as an external service but different matching algorithms could be supported.

4.1.2. Managing new experiments

As already discussed in Section 2, new experiments could be imported from different sources. Scientific and scholarly repositories represent a valuable resource for data already extracted from publications and stored in semi-structured formats (e.g., XML, YAML, or other custom formats). In other cases new experiments could be directly extracted by domain experts and inserted manually from supplementary material attached to publications or from data and plots included in paper themselves.

In this phase it is necessary to guarantee consistency, uniqueness and quality for all the acquired information avoiding missing data and acquisition-related errors. However, validation brings some challenges. Indeed, if on one side validation routines and quality checks are necessary to avoid input errors and partial information, on the other side an experiment type is not characterized by a fixed set of fields. Even if scientific experiments follow well codified and standardized procedures, as their description in the scientific literature, many variations could occur. For example some values could be described at a different aggregation level (e.g., indicating an average temperature instead of a temperature which changes slightly during the experiment), some papers could describe more experimental measurements

than others, some values could be missing in a paper because there exists a "standard" or "default" value (e.g., the atmospheric pressure) and there could be different sets of values which bring the same information (e.g., a time and a distance rather than a flow velocity). In general, it does not exists a pre-defined level of resolution to describe a scientific experiment, but it does exists a sufficient set of conditions that describe an experiment well enough to be understandable by a domain expert and reproducible with enough precision in a real setting or, as in our case, reproducible with enough precision through an automatic simulation.

The knowledge required to interpret and understand an experiment and the set of conditions needed to effectively simulate it are complex and domain specific. Moreover, they could evolve as the model evolves or some assumptions change. Therefore, input validation ensures the absence of inconsistencies (e.g., the usage of the wrong units for a certain variable), but the actual interpretation of the experiments should be dynamic and postponed. Similarly, in the acquisition phase data should never be modified and an experiment should simply be kept in the database without changes, even if the knowledge to automatically understand it is not fully available yet. There are two main reasons for not modifying experimental data in their original description:

- Assuming an experiment has no input-related errors, it is a "source of truth" by itself and the knowledge to automatically understand and simulate it could be available successively.
- This allows to externally and dynamically define a "domain knowledge" to interpret, change (aggregate, convert) and use experimental data for simulation and analysis tasks.

Note that this kind of validation, even the dynamic model-driven one, only concern the completeness, reproducibility, coherency and understandability of an experiment in terms of its "schema", that is the set of conditions and variables included, but not the reliability of its data. The latter requires the comparison of the experiment to other experiments/simulations and being able to correctly interpret it is a prerequisite.

4.1.3. Exploratory and cross analysis

Experiments and models in the framework can be searched and filtered by a generic set of fields. This allows selecting only those experiments/models which satisfy some conditions and executing cross analyses on them, enabling the discovery of behaviors and patterns which can not be derived from the analysis of a single instance, and allows handling volumes of data largely beyond those manageable manually.

Two main tasks in this context are *model validation* and *experiment validation*, which aim to validate the quality of a model or an experiment through large-scale comparisons and aggregated values, but also generic *data analysis* processing should be supported. All these tasks build on the stored experiments and models and on the automatic simulation capabilities of the framework, but also on a domain knowledge which the framework integrates to perform semantic-aware analysis and enhance the results.

These analysis directions are detailed in the following:

- *Model validation* allows assessing the *global* performance of a model in specific conditions. Once some conditions are expressed through a query, the model can be evaluated with respect to all the experiments satisfying the query and, optionally, with respect to other models for the same query set. Given the fact that each model-experiment pair brings to an *index* which expresses the performance of the model on the experiment (i.e., how much they match), such large-scale validation requires *data aggregation* to provide one or few global indeces for each model, e.g., a global index for each output variable, thus bringing to global performance indicators for the model.

- *Experiment validation* allows assessing the reliability of experimental data through large-scale cross-comparisons. Indeed, it has been recently highlighted that correlations which exist among similar experiments allow identifying inconsistent data [27] and the same approach can be extended comparing experiments to model outputs.
- Finally, generic *data analysis* functions need to be supported. In this regard, the framework needs to be extensible and the functions can be assembled in *workflows* to further analyze and aggregate simulation results. Examples of analysis in this respect are *outlier detection* to discover the experiments for which a model is less accurate, *clustering* to discover classes of experiments that bring to inconsistent results and *correlations* of model results with various (meta)data, such as *scholarly data* to examine the impact of experiments published by a certain author or journal.

All these analyses should be built around the concept of *exploratory computing* [20], which involves an iterative and multi-step process where results are summarized and visualized, iteratively refining the initial query.

4.1.4. Managing changes in models

Whenever a model is modified it could simply be considered a totally new and independent instance and validated against the whole set of stored experiments computing its validation indices. However, such an approach has at least two major shortcomings: lack of development tracking and required computational resources.

Development tracking. In almost all cases, a new model is a modified (potentially improved) version of an existing model along the development process already shown in Figure 1. This means that improvements should be expressed not only in absolute terms, but also, and more importantly, in relative terms with respect to the starting point. This reflects the fact that a change in a model has the primary goal to improve its performance. Therefore, considering a new model as a variation of an existing one helps tracking the relative changes. Doing so, the continuous "test and refinement" cycle leads to a *sequence of models* and for each pair of models of this sequence it should be easy to derive both the relative change in the model and the relative change in the output performance, allowing, ultimately, to link model changes to performance variations. In some cases, the user could have more of an exploratory intent. There could be the need of parametrizing an initial model and simultaneously testing different combinations to analyze the impact of one or more model hyper-parameter(s), in some ranges, on the resulting performance. This kind of cases is strongly related to the already seen one. Indeed, there is a *set* of new models and each one is tracked as a variation of the starting one and, at the same time, parallel to the other "branches".

Computational resources. As mentioned above, testing a model on an experiment requires to build a "simulation input" for the model, to execute it and to compare the outputs from the model with the experimental outputs. In the majority of cases, the actual model execution is the most time and resources demanding task, as typically happens in scientific workflows. Anyway, independently from the actual time/resources needed to simulate a model on a single experiment, bottlenecks could arise following the development cycle and continuously re-testing modified versions of the model on the whole set of experimental data. Considering a new model as a variation of an existing one helps in this, because a changes in the model could affect only *a portion* of its behavior, which, in turn, could affect only a subset of the whole experimental dataset, thus significantly reducing the number of executions needed. Moreover, from a more technical point of view, the re-execution of a slightly changed model could

benefit from intermediate results already obtained executing the starting one, thus reducing time and resources needed.

Therefore, it becomes crucial to follow model development and keeping track of versions and derivations. Once a model changes, it is possible to re-validate it only on the portion of experiments affected by the change.

4.2. A first extended prototype

A working prototype has been already developed and deployed. It has been completely re-written with respect to the initial prototype reported in [44] to take into account the limitations and the requirements outlined in that paper since from the very beginning.

The new prototype has been developed to better analyze existing and emerging use cases and requirements, and therefore it followed an *agile* development approach. For this reason it already includes a set of fully working features and some partially developed but anyway working ones. Much effort was spent in developing the integrated infrastructure and in the abstraction and integration layers which will allow further development and the evolution of this prototype in a full framework. The architecture, to which this prototype totally complies, is described in Section 6.

The client application has been developed as a web application to minimize client-side requirements. The server offers all its services through a REST API in the JSON format, allowing an easy integration with other applications and the possibility to develop alternative clients. Examples of JSON responses are shown in Figure 2. User authentication is supported, differentiating among services offered.

The prototype has been implemented using PostgreSQL[5] as operational database, using its extensions to support array data and semi-structured XML and JSON data. The backend has been written in Python, using the Django[6] framework as web framework with the Djangoi-REST[7] extension for the REST API. Python libraries used in the backend include: `numPy`, `sciPy`, `pandas`, `scikit` and `statistics` for data elaboration, `pint` to manage physical quantities and conversions, `subprocess` to manage parallel executions. The frontend is based on the React[8] framework and Plotly[9] has been used as graphing library.

The prototype has interfaces towards the Opensmoke [17] simulator to run models and to the Curve Matching algorithm [6] to perform experiment/model comparisons. Moreover, it has interfaces to import data through the ReSpecth [50] and the ChemKED formats [53]. New experiments can be also added by users through a dedicated interface using human-readable formats and interactive forms. At the moment, the prototype stores more than 1200 experiments. They include those available in the ReSpecth resposi-tory[10] and experiments manually uploaded, including the collection published within [27]. The number of experiments stored is growing.

The understanding of experimental data leverages on an external ontology, which is populated by domain experts and, in the current prototype, is built starting from `csv` human readable files. It allows to perform data cleaning, perform types and units conversions, correctly build the simulation input for the experiment and interpret output variables. Experiments can be stored without semantic constraints as

[5] https://www.postgresql.org/
[6] https://www.djangoproject.com/
[7] https://www.django-rest-framework.org
[8] https://reactjs.org/
[9] https://github.com/plotly/plotly.js/
[10] http://respecth.hu/

```
{
  - data: [
      - {
            model: "polimi1412",
            average_index: 0.8634703,
            average_error: 0.0027774,
            CO2_index: "0.8057320",
            CO2_error: "0.0041761",
            CO_index: "0.8559380",
            CO_error: "0.0015967",
            C2H5OH_index: "0.9287410",
            C2H5OH_error: "0.0025594"
        },
      - {
            model: "polimi1800",
            average_index: 0.877474,
            average_error: 0.0032408,
            CO2_index: "0.8177820",
            CO2_error: "0.0054314",
            CO_index: "0.8670340",
            CO_error: "0.0013645",
            C2H5OH_index: "0.9476060",
            C2H5OH_error: "0.0029266"
        },
      - {
            model: "polimi1809",
            average_index: 0.8777867,
            average_error: 0.0034936,
            CO2_index: "0.8198380",
            CO2_error: "0.0062727",
            CO_index: "0.8659550",
            CO_error: "0.0012850",
            C2H5OH_index: "0.9475670",
            C2H5OH_error: "0.0029230"
        }
    ],
  - names: [
        "CO",
        "CO2",
        "C2H5OH"
    ]
}
```

```
[
  - {
      id: 1902,
      reactor: "stirred reactor",
      experiment_type: "jet stirred reactor measurement",
      fileDOI: "10.24388/x00003001",
    - common_properties: [
      - {
            id: 2337,
            name: "pressure",
            units: "atm",
            value: "1.0000000000",
            sourcetype: null,
            experiment: 1902
        },
      - {
            id: 2338,
            name: "volume",
            units: "cm3",
            value: "30.0000000000",
            sourcetype: null,
            experiment: 1902
        },
      - {
            id: 2339,
            name: "residence time",
            units: "s",
            value: "0.0700000000",
            sourcetype: null,
            experiment: 1902
        }
    ],
    - initial_species: [
      - {
            id: 3665,
            name: "N2",
            units: "mole fraction",
            amount: "0.9740000000",
            cas: null,
            experiment: 1902
        },
      - {
            id: 3664,
            name: "O2",
            units: "mole fraction",
            amount: "0.0240000000",
            cas: null,
            experiment: 1902
```


Fig. 2. Examples of JSON responses from the REST interface. On the left, validation results are returned for a set of requested models and experiments. On the right, the list of experiments satisfying a certain query is returned and each experiment includes its metadata, boundary conditions and actual data (this response is not complete in the figure).

long as they satisfy schema constraints, but their semantic interpretation, which happens dynamically, is then required for their usage in the framework. At the moment, about one third of the stored experiments are interpretable by the prototype, but this number is growing rapidly as new domain knowledge is added.

Some screenshots taken from the extended prototype are shown in the following. Figure 3 shows the search form through which it is possible to filter experiments based on a query which includes a generic boolean condition and to select them, Figure 4 shows the validation interface through which it is possible to select a list of models to validate with respect to the experiments previously filtered and Figure 5 shows the *summary results* of this validation consisting in an average *index* in the range $[0, 1]$ for each model and each output variable. From there, results can be detailed, as shown in Figure 6. Detailed results include experimental/model curves and the validation index for each experiment, model and output variable.

This "search & execute" pipeline is supported by the prototype also for external functions. It is possible to easily define new functions which, given a set of experiments filtered through the generic conditions and taken into account other inputs (e.g. a list of models) and/or results (e.g. validation results), can compute new summary and detailed results for the set.

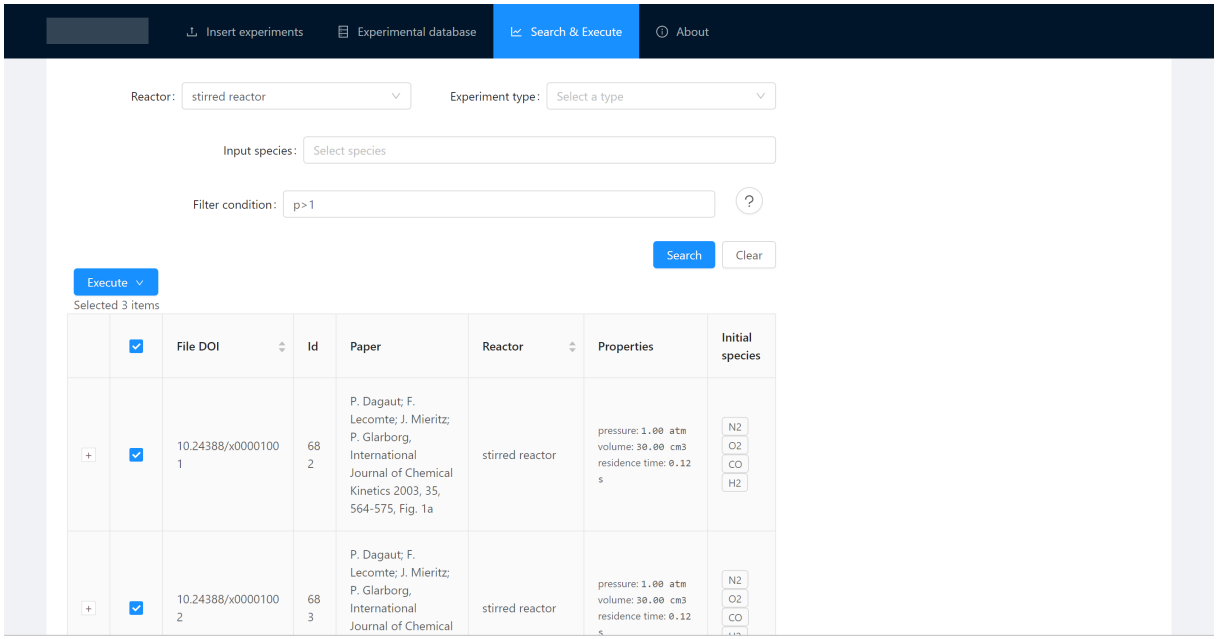

Fig. 3. Prototype/1. Search experiments by complex filter conditions and select the resulting subset.

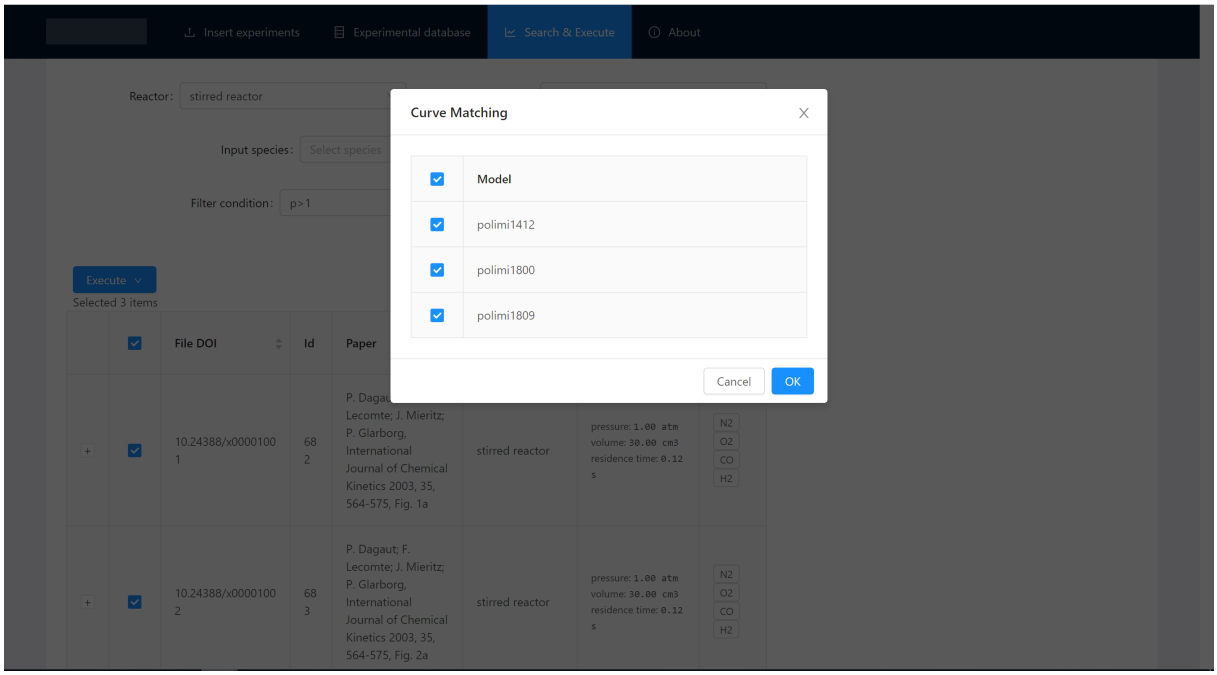

Fig. 4. Prototype/2. Select the models to be validated and compared against the set of experiments previously selected.

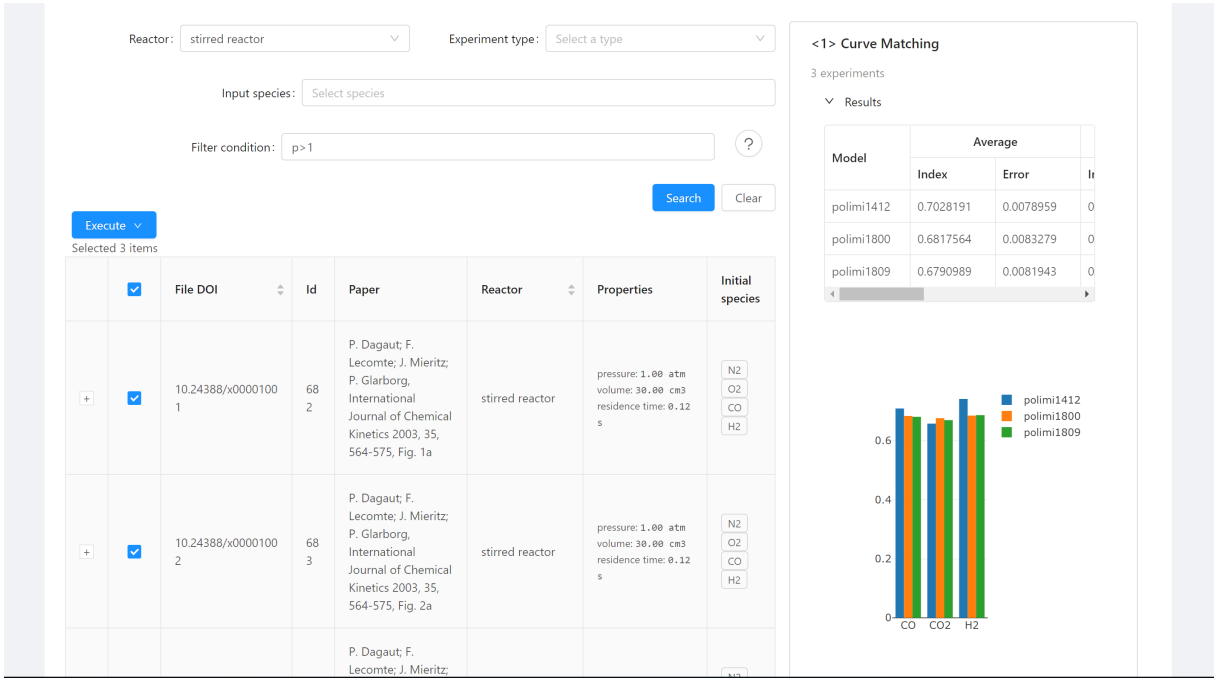

Fig. 5. Prototype/3. Aggregated results are shown for all the experiments and the models selected. An aggregated index in the range $[0, 1]$ is computed for each output variable and for each model and shown also as a bar chart.

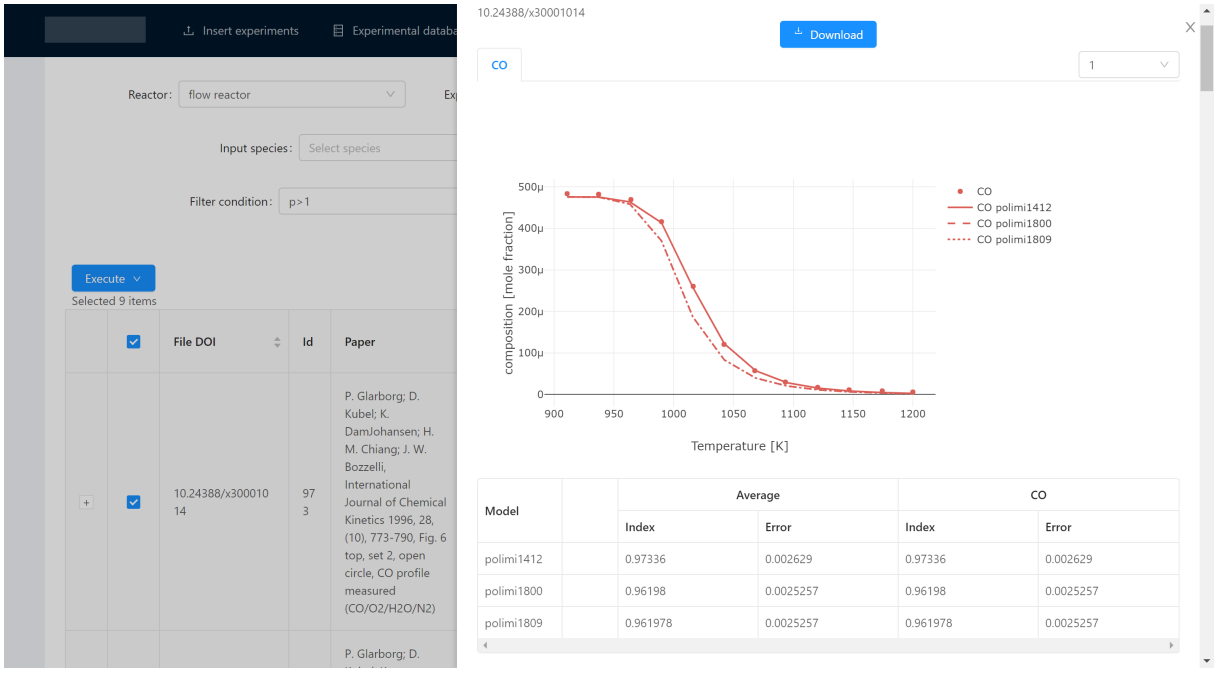

Fig. 6. Prototype/4. Detailed results can be shown upon request; in this case the index in the range $[0, 1]$ is shown for each model/experiment comparison, together with a chart reporting raw experimental outputs (dots) and the related simulated outputs for each model (lines).

5. New requirements

Starting from the scenario presented in Section 2 and the use cases described together with their challenges and the first prototype in Section 4, a new set of requirements have been formulated. The goal of these requirements is to enhance the efficiency and the effectiveness of data management and enable new exploration and analysis directions in this context. They are presented and discussed in the following, and will also drive the design of the new architecture presented in Section 6.

5.1. Continuous multi-source integration and quality management

The need for a continuous multi-source integration comes from the variability of the information sources, which could vary over time and cannot, therefore, be assumed a priori. This integration has many facets. First of all the *format* and the *structuredness* of the data varies. A *semantic* integration is necessary, since the same concept could be described differently in different instances, and this requires an ontology-based semantic layer and a dynamic interpretation of the information already stored. The information conveyed by the data is heterogeneous and can largely vary in terms of *accuracy*, *precision* and *coverage*. A continuous management of the already-acquired information is necessary in order to re-interpret the data already stored according to new knowledge, that could also derive from different sources.

Such dynamic environment impacts on the *information quality* (IQ) management: as new data and metadata are acquired or generated through processing, the IQ — with each single dimension that defines it — evolves. Stored objects do not only change over time, enriching their information, but are also characterized by explicit (*complex networks*) or implicit (*articulated objects*) interdependencies. For example, an experiment stred in the framework may have a certain quality which depends on its data, but further acquired "similar" experiments and their analysis could lead to the discovery of inconsistencies in the first experiment, affecting its quality, without changes in the experimental data itself. The IQ of such articulated objects is a function of the information quality of the sub-objects and of the other objects for which a relationship exists. Indeed, besides managing the quality of raw data, which is a problem addressed in the literature, the focus is on introducing a way of managing the quality of complex information through their relationships. Data dependencies over these relationships are in general not simply additive and certain characteristics of an object (an experiment or a model) emerge only through large-scale comparisons.

Knowledge extraction from complex relationships can benefit from the availability of domain ontologies as data sources and their integration in the analysis. An open issue is represented by the lack of a complete domain ontology. To face this challenge, a solution could be the automatic generation of ontological relationships based on the acquired data and *data mining* techniques [23, 47].

5.2. Continuous dynamic validation

The process of continuous validation entails the matching of already stored information with new information as it is acquired. This requirement mainly comes from the need of validating *models* and *experimental data*, one respect to the others through curve matching techniques, as when new experiments are acquired or when a model evolves, as described in Section 4.

Validation is performed through *cross-comparisons* which require efficient means for *extracting* the right data, *comparing* them taking into account the differences, the lacks and the uncertainties that could exist in their representations and *enriching* the objects (models and experimental data) with the results.

To do this, multi-source integration is a pre-requirement, especially to extract the right data. For example, to extract the experiments that need to be re-validated when a model evolves, it is necessary to assess the change in the model (model data), derive the set of experimental conditions affected (domain ontology) and extract the experiments which satisfy these conditions (experimental data).

Validation requires also to investigate the *reason* for some results obtained, since often there are many different situations that must be taken into consideration. For example, if some models fail in simulating a new experiment:

- The experiment could have some information which is missing in its original description, or missing information could derive from partial acquisition of experiment information.
- The experiment could have been acquired with all the necessary information, but its interpretation by the framework is not complete and brings to wrong simulations and comparisons.
- The experiment could be actually imprecise/unreliable.
- The models are not able to precisely simulate the condition described in the experiment because something is missing or imprecise in their design.

The consequences are very different: if the first two cases should be addressed modifying existing experimental or knowledge-related data in the framework, the third case requires to label an experiment as "not reliable", while the latter case should drive further testing and model refinement. More data can help the identification of the right case.

5.3. Dynamic acquisition

Large-scale validation and analysis tasks benefit from a large amount of experimental data. Even if data input is largely a semi-automated task, especially for experiments distributed in unstructured formats, more structured or complementary information (e.g., scholarly data) can be retrieved automatically from external sources. Moreover, the discovery of new sources of useful information (publications) can be automated.

This can be accomplished by a dynamic acquisition driven by the data already stored. The goal is to enhance the IQ of these data and improve data coverage by acquiring new information (for example, new experiments to benchmark existing models). This can be obtained by extending the concept of "focused crawling" [57]. The best predicate for querying new information from external sources, in general, changes over time and depends on a background knowledge. Acquired data can drive the acquisition of new data in a *virtuous cycle*. The background knowledge is certainly composed by the already acquired information, but also by the list of preceding queries and their results. Indeed, when acquiring data from unreliable sources it is not possible to make strong assumptions about them and an *exploratory approach* must be employed (see Subsection 5.4).

5.4. Data exploration

Articulated (meta)data can provide support for interactive and evolving data exploration and *exploratory computing* [20]. This has to do with the need of automatically or manually querying for information which are in general incomplete, heterogeneous or may not exist at all. Moreover, manual interventions are key to maintain an overall high IQ resolving conflicts and enriching domain knowledge, and they need to be supported by effective query techniques. These include summarization [14], result refinement, iterative exploration [52] and a *top-k* query processing environment [46], thus improving the returned "manageable" results.

6. Proposed architecture

In this section an architecture for the framework is proposed and outlined. Starting from the functional requirements discussed previously, this section will focus primarily on the *architectural requirements*. A detailed architectural design, together with its development and testing, are out of the scope of this paper and represent ongoing work started with this analysis. As illustrated in Subsection 4.2, some of the components have been already developed in the prototype.

In the following, Subsection 6.1 describes the data model, while Subsection 6.2 gives an overview of the global architecture.

6.1. Data model

In this subsection the data model for the proposed architecture is discussed.

The system requirements discussed in Section 5 translate into a set of data-related requirements:

- Supporting *continuous data integration* to combine data from different sources, including external sources, without assumptions on their number and type (structured and semi-structured) and allowing each individual source to evolve independently and dynamically handling data and format conflicts.
- *Analytical querying* to support the various kinds of analysis, which are in general expressed at high level and inherently combine various sources of information at an aggregated level.
- Aiding the *exploration* and the *discovery* of relevant data through user-defined ontologies and reasoning capabilities.

Most of these requirements are normally addressed through DW and OLAP systems, which support the analytical querying of an integrated set of data sources. However, traditional OLAP systems do not fit all these requirements. Indeed, in a traditional OLAP system all the data sources should be known and described in advance, as the rules used to perform their integration, and a traditional ETL process does not allow a continuous integration, especially when there are external sources. Moreover, traditional OLAP systems typically do not deal with semi-structured data and external ontologies.

*Exploratory OLAP*s supported by semantic technologies can overcome these limitations [2] and their "open-world" assumption allows analytical querying of new external data.

Several data sources contribute to the analysis tasks:

- The *operational database*, which stores experimental data, models (as logical paths, without internal information) and the results of the simulations and the analysis performed by external tools. This is a relational database continuously fed by the framework.
- Internal structure of *scientific models*, to extract detailed information about them, perform comparisons and keep track of the development workflow. A scientific model, besides being "executed" by the simulator, can be also seen as a data source, since its internal information can give hints about its results and this allows to link changes in its internal structure to changes in its performance.
- *Domain ontologies*, which support the integration, the exploration and the analysis of all the data and evolve independently and externally with respect to the framework. These are semi-structured linked data.
- *Scholarly data*, which can be the source of new experimental data and aid the integration, exploration and analysis tasks.

Each of these sources is discussed in the following.

Operational database. The operational database is the base for simulations and analysis tasks. Its main goal is to uniquely identify each experiment, model, simulation and analysis outcome. This guarantees consistency and avoids the re-execution of the same model on the same experiment and allows re-using results from intermediate analysis. The operational database has been designed as an extended relational database to privilege performance and ensure the same schema for all the stored data. It does not include most of the data not strictly needed to simulate a given experiment with a certain model. Therefore, it stores experimental data but not detailed scholarly metadata and it stores the model "logical path", necessary to execute it, but not its internal information. This allows having a compact and regular schema and relying on the dynamic integration of external and potentially less structured data sources for more complex analysis.

Scientific model. A kinetic model describes the network of reactions, and the relative rate at which these reactions proceed, through which a reactant, or a mixture of reactants, is converted into products. While many disciplines dealing with kinetic modelling (e.g., catalytic processes) strongly rely on largely empirical and simplified models, the relative simplicity of gas phase combustion allowed to gain an extremely high level of detail thus resulting in complex kinetic models involving $10^2 - 10^3$ chemical species and $10^3 - 10^4$ elementary reaction steps, plus thermodynamic parameters and transport properties. Overall, the number of strongly interconnected model parameters can be as large as $10^5 - 10^6$. From an operational point of view a kinetic model is a structured file. It could be translated in the future into a graph data model, such as RDF, to be queried. Models evolve externally with respect to the framework and accessing their internal data enables some analyses which are not feasible considering them as black boxes, e.g., linking model internal changes to performance improvements.

Domain ontologies. Domain ontologies are needed to interpret experimental data and results and enable their integration, exploration and analysis. For example, they describe which subsets of boundary conditions are semantically equivalent and can be converted from one to another, which boundary conditions should be associated with an experiment to be reproducible with the simulator, which are the output variables that need to be validated for each experiment type, and so on. An ontology is managed as an external data source because it can be imported from existing open data and, even if created on purpose for the framework when such information is not already available, it is managed outside the framework. As a complete domain ontology does exists at present, semi-automated generation techniques [56] or data mining techniques on existing data [23, 47] can be used for the generation of the information needed.

Scholarly data. Since experiments are normally extracted from scientific publications and each experiment keeps track of its original source, the integration of a scholarly data repository can aid many of the identified requirements. First of all, this integration avoids the manual input of detailed scholarly data while inserting a new experiment, since the experiment can just be associated with the publication identifier and the related information automatically retrieved. This avoids inconsistencies which could arise inputting and storing this kind of information as more or less structured text, as currently happens in experimental repositories in this domain [22, 26, 50, 53], such as naming ambiguities and missing data. The integration of scholarly data and *citation network analysis* techniques on them can aid the discovery of new and potentially relevant publications to acquire experiments based on the publications which are the sources of the experiments already in the operational database. Detailed and structured scholarly data support exploration and enable scholarly-related data aggregation, for example to aggregate analysis results for experiments published by the same author or journal. Currently, many open scholarly data repositories and analysis tools are available [21, 55], as initiatives to publish open bibliographic citation information as linked data, such as Open Citations [39].

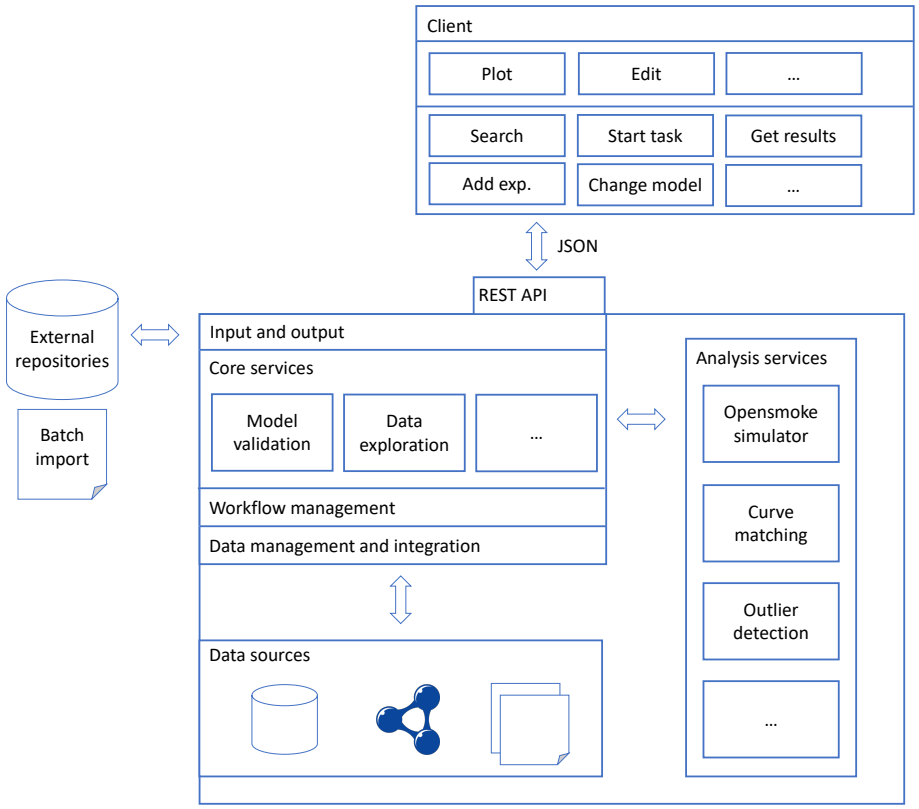

Fig. 7. Sketch of the architecture for the framework.

6.2. *Framework*

The main architectural components are sketched in Figure 7.

A central server includes all the data sources, the core services, the input and output layer, external analysis services, workflow management and data management. The central server is largely designed following a service-oriented architecture (SOA). A service-based approach has two advantages: the integration of existing components with new components and the autonomous development of each of the services.

The *core services* provide the core functions of the framework: model validation, data and results exploration, and so on. These services are designed to interface with analysis services on one side, which provide actual simulation and computational results, and with the data management and integration layer on the other side, which provides access to the data sources.

The *data management and integration* layer handles the different data sources, data cleaning and quality management. It allows the core services to perform complex, exploratory and quality-aware queries over them. This layer should be able to efficiently search, retrieve and integrate data based on the patterns of the acquisition/validation/exploration cycle. For example, retrieve all the experiments needed to validate a certain model change.

The *analysis services* provide a wide range of functions to compute validation indices, statistical analysis, aggregated values, and so on. Model simulation is managed as an analysis service. Each function needs to be wrapped in a service standardizing its interface. This allows their uniform management as generic components and the storage of their outputs with an integrated schema. Each analysis outcome is stored to be reused as needed, even as sub-step in a more articulated task or by a different user. This kind of services is designed to be developed easily by third parties or starting from existing components.

A *task* in the framework is defined over the core services which, in turn, refer to the data management layer or to analysis services for specific sub-tasks. The *workflow management* architectural component handles functions such as resource allocation, task scheduling and parallelization. This component could be an existing workflow management system (see Section 3) integrated in framework.

Given the variety of existing formats and standards, the *input and output* layer should integrate conversion tools. The output interface allows accessing both experimental/model data and analysis outcomes, while the input interface allows inserting or modifying experiments and models. On top of the input/output layer, a REST API allows interacting with the framework and exporting data as JSON.

The web client accesses the server through the REST API and provides a user friendly way of interacting with the framework. This includes searching, starting a task, browse the results, add experiments or make changes in models. Moreover, it includes additional client-side functions, such as plots and graphs.

7. Concluding remarks

The effective integration of large-scale scientific data analysis in the development cycle of scientific models, especially in their validation, is one of today's challenges, given the increasing availability of scientific open data and tools to automatically assess model's performance, potentially allowing rapid and extremely significant technological advancements. Starting from the well defined domain of combustion kinetic modelling, this work takes a more general perspective analyzing the use cases and the emerging requirements for an integrated framework which aims to automatically leverage large-scale scientific data analysis in the scientific development cycle, especially in the validation phase. A preliminary architecture has been designed, and an extended prototype has been deployed, setting the basis for its refinement and development in future activities. This work mainly focused on data-related requirements, since data integration, interpretation and validation are a pre-requisite for every subsequent analysis task. Future work includes the integration of the proposed framework with other techniques in the model development cycle. In particular, an effective integration with automatic model development techniques, which constitute a recent and very promising direction [15, 16], could highlight critical conditions that need further refinement in these models and bring to an automatic development and validation cycle.

References

[1] L. Aagesen, J. Adams, J. Allison, W. Andrews, V. Araullo-Peters, T. Berman, Z. Chen, S. Daly, S. Das, S. DeWitt, et al. Prisms: An integrated, open-source framework for accelerating predictive structural materials science. *JOM*, 70(10):2298–2314, 2018.

[2] A. Abelló, O. Romero, T. B. Pedersen, R. Berlanga, V. Nebot, M. J. Aramburu, and A. Simitsis. Using semantic web technologies for exploratory OLAP: a survey. *IEEE Transactions on Knowledge and Data Engineering*, 27(2):571–588, 2015.

[3] D. Ardagna, C. Cappiello, W. Samá, and M. Vitali. Context-aware data quality assessment for big data. *Future Generation Computer Systems*, 89:548–562, 2018.

[4] M. Atkinson, S. Gesing, J. Montagnat, and I. Taylor. Scientific workflows: Past, present and future. *Future Generation Computer Systems*, 75:216 – 227, 2017.
[5] J. M. Bergthorson and M. J. Thomson. A review of the combustion and emissions properties of advanced transportation biofuels and their impact on existing and future engines. *Renewable and sustainable energy reviews*, 42:1393–1417, 2015.
[6] M. Bernardi, M. Pelucchi, A. Stagni, L. Sangalli, A. Cuoci, A. Frassoldati, P. Secchi, and T. Faravelli. Curve matching, a generalized framework for models/experiments comparison: An application to n-heptane combustion kinetic mechanisms. *Combustion and Flame*, 168:186–203, 2016.
[7] P. L. Bhoorasingh, B. L. Slakman, F. Seyedzadeh Khanshan, J. Y. Cain, and R. H. West. Automated transition state theory calculations for high-throughput kinetics. *The Journal of Physical Chemistry A*, 121(37):6896–6904, 2017.
[8] C. L. Borgman. *Big data, little data, no data: Scholarship in the networked world*. MIT Press, 2015.
[9] D. Calvanese, B. Cogrel, S. Komla-Ebri, R. Kontchakov, D. Lanti, M. Rezk, M. Rodriguez-Muro, and G. Xiao. Ontop: Answering SPARQL queries over relational databases. *Semantic Web*, 8(3):471–487, 2017.
[10] C. Cappiello, W. Samá, and M. Vitali. Quality awareness for a successful big data exploitation. In *Proceedings of the 22nd International Database Engineering & Applications Symposium*, pages 37–44. ACM, 2018.
[11] C. Cavallotti, M. Pelucchi, and S. Klippenstein. EStokTP: Electronic structure to temperature and pressure dependent rate constants. *unpublished*, 2017.
[12] A. Chalamalla, I. F. Ilyas, M. Ouzzani, and P. Papotti. Descriptive and prescriptive data cleaning. In *Proceedings of the 2014 ACM SIGMOD international conference on Management of data*, pages 445–456. ACM, 2014.
[13] X. Chu, I. F. Ilyas, S. Krishnan, and J. Wang. Data cleaning: Overview and emerging challenges. In *Proceedings of the 2016 International Conference on Management of Data*, pages 2201–2206. ACM, 2016.
[14] A. Cohan and N. Goharian. Scientific article summarization using citation-context and article's discourse structure. *arXiv preprint arXiv:1704.06619*, 2017.
[15] C. W. Coley, R. Barzilay, T. S. Jaakkola, W. H. Green, and K. F. Jensen. Prediction of organic reaction outcomes using machine learning. *ACS central science*, 3(5):434–443, 2017.
[16] C. W. Coley, W. H. Green, and K. F. Jensen. Machine learning in computer-aided synthesis planning. *Accounts of chemical research*, 51(5):1281–1289, 2018.
[17] A. Cuoci, A. Frassoldati, T. Faravelli, and E. Ranzi. Opensmoke++: An object-oriented framework for the numerical modeling of reactive systems with detailed kinetic mechanisms. *Computer Physics Communications*, 192:237–264, 2015.
[18] A. de Waard. Research data management at Elsevier: Supporting networks of data and workflows. *Information Services & Use*, 36(1-2):49–55, 2016.
[19] E. Deelman, T. Peterka, I. Altintas, C. D. Carothers, K. K. van Dam, K. Moreland, M. Parashar, L. Ramakrishnan, M. Taufer, and J. Vetter. The future of scientific workflows. *The International Journal of High Performance Computing Applications*, 32(1):159–175, 2018.
[20] N. Di Blas, M. Mazuran, P. Paolini, E. Quintarelli, and L. Tanca. Exploratory computing: a comprehensive approach to data sensemaking. *International Journal of Data Science and Analytics*, 3(1):61–77, 2017.
[21] C. Dunne, B. Shneiderman, R. Gove, J. Klavans, and B. Dorr. Rapid understanding of scientific paper collections: Integrating statistics, text analytics, and visualization. *Journal of the American Society for Information Science and Technology*, 63(12):2351–2369, 2012.
[22] M. Frenklach. Transforming data into knowledge - Process informatics for combustion chemistry. *Proceedings of the combustion Institute*, 31(1):125–140, 2007.
[23] K. Gábor, H. Zargayouna, I. Tellier, D. Buscaldi, and T. Charnois. A typology of semantic relations dedicated to scientific literature analysis. In *International Workshop on Semantic, Analytics, Visualization*, pages 26–32. Springer, 2016.
[24] C. W. Gao, J. W. Allen, W. H. Green, and R. H. West. Reaction mechanism generator: Automatic construction of chemical kinetic mechanisms. *Computer Physics Communications*, 203:212–225, 2016.
[25] H. Gossler, L. Maier, S. Angeli, S. Tischer, and O. Deutschmann. Carmen: a tool for analysing and deriving kinetics in the real world. *Physical Chemistry Chemical Physics*, 20(16):10857–10876, 2018.
[26] G. L. Goteng, N. Nettyam, and S. M. Sarathy. Cloudflame: Cyberinfrastructure for combustion research. In *Information Science and Cloud Computing Companion (ISCC-C), 2013 International Conference on*, pages 294–299. IEEE, 2013.
[27] N. Hansen, X. He, R. Griggs, and K. Moshammer. Knowledge generation through data research: New validation targets for the refinement of kinetic mechanisms. *Proceedings of the Combustion Institute*, 2018.
[28] H. Jagadish. Big data and science: Myths and reality. *Big Data Research*, 2(2):49–52, 2015.
[29] M. Keçeli, Y.-P. L. Elliott, M. Johnson, C. Cavallotti, Y. Georgievskii, W. P. Green, J. Wozniak, A. M. Jasper, and S. Klippenstein. Automated computational thermochemistry for butane oxidation: A prelude to predictive automated combustion kinetic. *Proceedings of the Combustion Institute, in press*, 2018.
[30] L. Libkin, W. Martens, and D. Vrgoč. Querying graphs with data. *Journal of the ACM (JACM)*, 63(2):14–53, 2016.
[31] C. S. Liew, M. P. Atkinson, M. Galea, T. F. Ang, P. Martin, and J. I. V. Hemert. Scientific workflows: Moving across paradigms. *ACM Computing Surveys (CSUR)*, 49(4):66, 2017.

[32] L. H. Marcial and B. M. Hemminger. Scientific data repositories on the web: An initial survey. *Journal of the American Society for Information Science and Technology*, 61(10):2029–2048, 2010.
[33] C. Olm, I. G. Zsély, R. Pálvölgyi, T. Varga, T. Nagy, H. J. Curran, and T. Turányi. Comparison of the performance of several recent hydrogen combustion mechanisms. *Combustion and Flame*, 161(9):2219–2234, 2014.
[34] F. Osborne, E. Motta, and P. Mulholland. Exploring scholarly data with rexplore. In *International semantic web conference*, pages 460–477. Springer, 2013.
[35] I. V. Pasquetto, B. M. Randles, and C. L. Borgman. On the reuse of scientific data. *Data Science Journal*, 16(8):1–9, 2017.
[36] I. V. Pasquetto, A. E. Sands, P. T. Darch, and C. L. Borgman. Open data in scientific settings: From policy to practice. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, pages 1585–1596. ACM, 2016.
[37] W. Pejpichestakul, E. Ranzi, M. Pelucchi, A. Frassoldati, A. Cuoci, A. Parente, and T. Faravelli. Examination of a soot model in premixed laminar flames at fuel-rich conditions. *Proceedings of the Combustion Institute, in press*, 2018.
[38] M. Pelucchi. Development of kinetic mechanisms for the combustion of renewable fuels. *PhD Thesis, Politecnico di Milano*, 2017.
[39] S. Peroni, D. Shotton, and F. Vitali. One year of the opencitations corpus. In *International Semantic Web Conference*, pages 184–192. Springer, 2017.
[40] E. Ranzi, A. Frassoldati, A. Stagni, M. Pelucchi, A. Cuoci, and T. Faravelli. Reduced kinetic schemes of complex reaction systems: fossil and biomass-derived transportation fuels. *International Journal of Chemical Kinetics*, 46(9):512–542, 2014.
[41] A. Rigamonti. Automatic Modeling System: a database based infrastructure to develop, validate and evaluate scientific models. An application to combustion kinetic models, Graduation Thesis, Politecnico di Milano, 2017.
[42] P. Ristoski and H. Paulheim. Semantic web in data mining and knowledge discovery: A comprehensive survey. *Web semantics: science, services and agents on the World Wide Web*, 36:1–22, 2016.
[43] M. A. Rodriguez and R. Buyya. A taxonomy and survey on scheduling algorithms for scientific workflows in iaas cloud computing environments. *Concurrency and Computation: Practice and Experience*, 29(8):e4041, 2017.
[44] G. Scalia, M. Pelucchi, A. Stagni, T. Faravelli, and B. Pernici. Storing combustion data experiments: New requirements emerging from a first prototype. In A. González-Beltrán, F. Osborne, S. Peroni, and S. Vahdati, editors, *Semantics, Analytics, Visualization, - 3rd International Workshop, SAVE-SD 2017, Perth, Australia, April 3, 2017, and 4th International Workshop, SAVE-SD 2018, Lyon, France, April 24, 2018, Revised Selected Papers, LNCS, volume 10959*, pages 138–149, Cham, 2018. Springer International Publishing.
[45] A. Schätzle, M. Przyjaciel-Zablocki, S. Skilevic, and G. Lausen. S2RDF: RDF querying with SPARQL on spark. *Proceedings of the VLDB Endowment*, 9(10):804–815, 2016.
[46] M. A. Soliman, I. F. Ilyas, and K. C.-C. Chang. Top-k query processing in uncertain databases. In *Data Engineering, 2007. ICDE 2007. IEEE 23rd International Conference on*, pages 896–905. IEEE, 2007.
[47] S. Spangler, A. D. Wilkins, B. J. Bachman, M. Nagarajan, T. Dayaram, P. Haas, S. Regenbogen, C. R. Pickering, A. Comer, J. N. Myers, et al. Automated hypothesis generation based on mining scientific literature. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1877–1886. ACM, 2014.
[48] A. Stagni, A. Frassoldati, A. Cuoci, T. Faravelli, and E. Ranzi. Skeletal mechanism reduction through species-targeted sensitivity analysis. *Combustion and Flame*, 163:382–393, 2016.
[49] R. Van de Vijver, K. M. Van Geem, and G. B. Marin. On-the-fly ab initio calculations toward accurate rate coefficients. *Proceedings of the Combustion Institute*, 2018.
[50] T. Varga, T. Turányi, E. Czinki, T. Furtenbacher, and A. Császár. Respecth: a joint reaction kinetics, spectroscopy, and thermochemistry information system. In *Proceedings of the 7th European Combustion Meeting*, volume 30, pages 1–5, 2015.
[51] H. Wang and D. A. Sheen. Combustion kinetic model uncertainty quantification, propagation and minimization. *Progress in Energy and Combustion Science*, 47:1–31, 2015.
[52] A. Wasay, M. Athanassoulis, and S. Idreos. Queriosity: Automated data exploration. In B. Carminati and L. Khan, editors, *2015 IEEE International Congress on Big Data, New York City, NY, USA, June 27 - July 2, 2015*, pages 716–719. IEEE, 2015.
[53] B. W. Weber and K. E. Niemeyer. ChemKED: A human-and machine-readable data standard for chemical kinetics experiments. *International Journal of Chemical Kinetics*, 50(3):135–148, March 2018.
[54] 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*, 3:160018:1–9, 2016.
[55] F. Xia, W. Wang, T. M. Bekele, and H. Liu. Big scholarly data: A survey. *IEEE Transactions on Big Data*, 3(1):18–35, 2017.
[56] Z. Xiang, J. Zheng, Y. Lin, and Y. He. Ontorat: automatic generation of new ontology terms, annotations, and axioms based on ontology design patterns. *Journal of biomedical semantics*, 6(1):4, 2015.

[57] R. Yu, U. Gadiraju, B. Fetahu, and S. Dietze. Adaptive focused crawling of linked data. In *International Conference on Web Information Systems Engineering*, pages 554–569. Springer, 2015.
[58] J. Zeng and K. W. Glaister. Value creation from big data: Looking inside the black box. *Strategic Organization*, 16(2):105–140, 2018.