Provenance in Collaborative in Silico Scientific Research: a Survey

Eduardo Jandre
Instituto de Computação
Universidade Federal Fluminense, UFF, Brazil
eduardojandre@id.uff.br

Bruna Diirr
Programa de Pós-Graduação em Informática
Universidade Federal do Estado do Rio de Janeiro, UNIRIO, Brazil
bruna.diirr@uniriotec.br

Vanessa Braganholo
Instituto de Computação
Universidade Federal Fluminense, UFF, Brazil
vanessa@ic.uff.br

ABSTRACT

Science is a collaborative activity by definition. Research is usually conducted by several scientists working together, and this behavior has been intensified in recent years. Furthermore, experiments are increasingly performed in silico, which demands proper support tools. Provenance-aware Workflow Management Systems and script-based tools have been popular ways of running in silico experiments, but these tools often neglect the collaboration aspect. Even solutions that aim at collaborative experiments do not always address the collaborators' needs. Literature shows surveys discussing subjects related to in silico experiments. However, they either focus on provenance collection and applications, thus treating collaboration as just another possible application, or focus on Workflow Management Systems, only listing collaboration as a possible challenge. This article surveys available tools and approaches that aim at aiding scientists to conduct collaborative in silico experiments. Particularly, we focus on challenges related to the provenance of these collaborative experiments. We devise a taxonomy with the aspects of collaboration in scientific research and discuss each of these aspects. We also identify literature gaps that provide future opportunities.

1. INTRODUCTION

Scientific knowledge is built incrementally and cumulatively. To discover something new, scientists have to extensively study their fields to understand the current state of the art. Additionally, an important part of the scientific process is the communication of the work done and the outcomes reached, which allows the scientific community to analyze and review other scientist's research and the obtained results. This process is essential because it allows other people to double-check the ideas, find flaws, or reproduce the achieved results, besides enabling the use of acquired knowledge in future discoveries [8]. Hence, collaboration plays a key role in scientific research and knowledge acquisition.

"Scientific collaboration can be defined as interaction taking place within a social context among two or more scientists that facilitates the sharing of meaning and completion of tasks with respect to a mutually shared, super-ordinate goal" [54]. Therefore, scientific collaboration occurs not only after the publication but especially in ongoing research. Research is usually carried out by several scientists working together. Indeed, collaboration is often encouraged and even required by research funding agencies [54].

Wuchty, Jones, and Uzzi [66] analyze almost 20 million publications from the mid-50s to the early 21st century, and conclude that the production of publications by teams of collaborators has increased over time and that these teams have grown in size. Also, the authors conclude that publications produced in teams usually receive more citations on average than publications made by a single author, even when self-citations are ignored [66].

At the same time, computer technology has advanced hugely. Computers have become cheaper and more accessible, and computer networks have spread all around the world. This movement produced two direct effects: (i) it allowed collaboration to occur not just between people nearby but also between people located all around the world; and (ii) it increased the number of scientific experiments conducted in silico.

In silico experiments typically demand more support from data management and software engineering tools when compared to other experiment classes (in vivo, in vitro, and in virtuo) [60]. Workflow Management Systems [3, 26, 67] and Script-based systems [17, 36, 47] (referred in this work as *Experiment Management Systems*) have been popular ways of running such experiments. However, collaboration is still one of the challenges in the area [16, 27, 31].

The data related to in silico experiments are not limited to the results of the experiment but also include the logical sequence of performed activities; parameters used; intermediary results of activities; information about the execution environment; etc. [25]. It is common for these data to be collected and stored in a provenance database. Provenance is a broad concept that can be applied in many disciplines and is usually linked to the origin of an object or data. It can be seen as a set of metadata that describes not only the object or data itself but also the activities applied in its production process. Bringing the concept into scientific research, it refers to information on how the experiment was performed and how the research results were recorded [31]. This should also include records of how the collaboration was conducted.

Provenance gathering is a common feature in many Experiment Management Systems [3, 17, 26, 31, 36, 47, 67]. However, when focusing on collaborative experiments, two challenges emerge: (C1) how to collect provenance in a collaborative experiment (this comprises collecting provenance of actions of scientists that may be working in different parts of the experiment or different geographical locations and machines); and (C2) how provenance can be used to make collaboration easier in this environment.

The main goal of this article is to map the state-of-the-art approaches and provenance-aware models that are available to conduct in silico collaborative experiments. We aim at investigating how they address challenges C1 and C2. To do so, we plan to answer the following research questions: (R1) How do existing tools store and collect provenance in a collaborative experiment?; (R2) how do existing tools use provenance to make collaboration easier in scientific experiments?. The research question R1 and R2 are respectively linked to challenges C1 and C2.

To answer these questions, we make a snowballing [30] based survey. We evaluate 170 publications and select 20 approaches and 7 surveys. To be selected, an approach has to satisfy the following criteria: (i) has collaboration as a focus (i.e., the problem to be solved or the subject of a survey); or (ii) has provenance as a focus while discussing collaboration features; and (iii) is in the context of in silico scientific experiments. The surveys were used to reinforce this work's motivation and as a benchmark. From the 20 selected approaches, 15 are tools for collaborative experiments, 2 are provenance-aware data models for collaborative experiments, and 3 approaches present both a tool and a provenance-aware data model for collaborative experiments.

This article contrasts with existing surveys [6, 16, 27, 31, 37, 51, 66] as follows. This work differs from Lu and Zhang's work [37] and Belloum et al. [6] by bringing a more detailed and up-to-date view of the work in the area. Besides that, Belloum et al. discuss the challenges to support e-science collaborative experiments with a closer look at the experiment life cycle, but it only addresses the tools provided by the VL-e project. Wuchty et al. [66] aim to demonstrate that teams have been increasingly dominating the scientific research in the production of knowledge, without addressing available tools and research that helps the execution of this type of experiment. On the other hand, Davidson and Freire [16] and Gil et al. [27] focus on the challenges and opportunities existing in the Workflow Management Systems research, without detailing the available tools. Other publications focus on provenance collection and its applications, and collaboration merely appears as one of the possible applications of provenance [31, 51]. As opposed to that, this survey focuses on provenance-related aspects of collaboration.

The article proceeds as follows: Section 2 presents an analysis of the existing provenance models that aim to precisely represent collaborative research; Section 3 discusses some aspects of collaborative research and proposes a taxonomy to capture the aspects that may influence collaboration in the scientific research scenario; Section 4 discusses publications and opportunities in the field; and Section 5 concludes the article.

2. PROVENANCE MODELS

Provenance is a broad concept and can be seen from different perspectives. Ragan et al. [51] classify provenance in five types: *Data provenance* (the history of changes and movement of data); *Visualization provenance* (the history of graphic views and visualization states); *Interaction provenance* (the history of user interaction with a system); *Insight provenance* (the history of cognitive outcomes and information derived from the analysis process); and *Rationale provenance* (the history of reasoning and intentions behind decisions, hypotheses, and interactions) [51].

Collaboration brings additional challenges in collecting and storing provenance. The first challenge (C1) resides in how to collect provenance in a collaborative experiment. It involves collecting data, interaction, and visualization provenance from multiple devices since scientists usually work on their workstation. Few initiatives capture provenance from multiple devices [18, 20, 64], but they usually

focus in high-performance settings, where a single user executes parts of the experiment in the cloud, cluster, or grid. This is different from having several scientists working on their local workstations, where there is usually no central control. Collecting this provenance could be useful in several situations, such as giving credit to those involved in the research [31], auditing the research, enabling the reproducibility of the experiment and providing relevant information that allows each member of a group to better understand the actions of other members in the context of a collaborative scientific experiment. Another challenge (C2) resides in how to use this provenance to make collaboration easier in a collaborative environment.

The first step to overcoming these challenges is providing a provenance model that can properly represent the research collaboration aspects. This model needs to represent four main aspects [37]: (i) Distribution (D) – Collaboration typically involves resources from multiple organizations; (ii) Heterogeneity (H) – Provenance produced by different workflows may have different formats. Even those that conform to the same schema may evolve during the experiment life cycle; (iii) Multilevel (M) – Experiments usually have complex tasks that are modeled hierarchically (e.g., using sub-workflows, or by functions calling functions in a script). Although this is not a specificity of collaborative experiments, the provenance model should store this hierarchy; (iv) Collaboration (C) – The model must support new user iterations and collaboration standards, besides storing information about these collaborations.

The term collaborative workflow has been used with multiple meanings in the literature. It is understood both as the *collaboration between workflows* or the *collaboration between workflow users* [37]. Collaboration between workflow users is the direct collaboration of users in the context of a scientific workflow. On the other hand, a collaboration between workflows refers to the indirect use of data produced by another workflow. This suggests an implicit collaboration, when collaboration occurs through the data published by another researcher.

Altintas et al. [1, 2] propose the provenance model shown in Figure 1, which is capable of capturing *implicit collaborations* within a scientific experiment. The model predicts the identification of workflows dependency from the relations between dataflows input and output, and also helps to identify contributions from users who collaborate on a project based on records of past executions. The authors extend OPM (Open Provenance Model) [44]

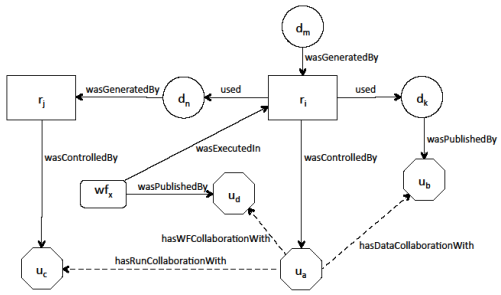

Figure 1: An abstract model of collaborative provenance nodes and dependencies using the extended Open Provenance Model [2]

to record user interactions when publishing data and workflows, which is essential for identifying the various types of user collaboration. This model explicitly represents *collaboration amongst users* (agents u_i in the figure) and which users were responsible for each run of the experiment (r_i in the figure). According to Ragan et al.'s classification [51], it captures *data* and *interaction* provenance. The approach also proposes a query language, which is an extension of the QLP (Query Language for Provenance) [5].

Missier et al. [43] propose a model that facilitates the sharing of provenance in collaborative environments. The model aims to provide end-to-end support for *implicit collaborations*. The approach treats sharing as an action from which provenance has to be preserved, i.e., the focus is to register the provenance of the data sharing process. To do so, the model adds new information to provenance traces, *stitching* common parts of those traces. With this, the model can represent cases when scientists use data that was produced by another scientist's workflow, even when they come from heterogeneous workflow systems. This model can represent *data* and *interaction* provenance [51].

Zhang et al. [68], Confucius [70], and ProvDB [41] present provenance models and tools that track collaboration provenance. Zhang et al. [68] propose the Collaborative Provenance Model (CPM), which is an extension of PROV-DM (PROV Data Model) [45]. Figure 2 shows that the model explicitly represents *Person* and *Group* of *Person* (a collaborating group), besides versions of *Workflow*, *Processor*, and *Data Links*. It also captures which user operates which workflow version, process version, and data link version. The model captures *data* and *interaction* provenance [51].

Confucius [70] introduces a provenance ontology (Figure 3). The ontology aims at supporting the capture and record of scientific workflow composition and user interactions during the process of

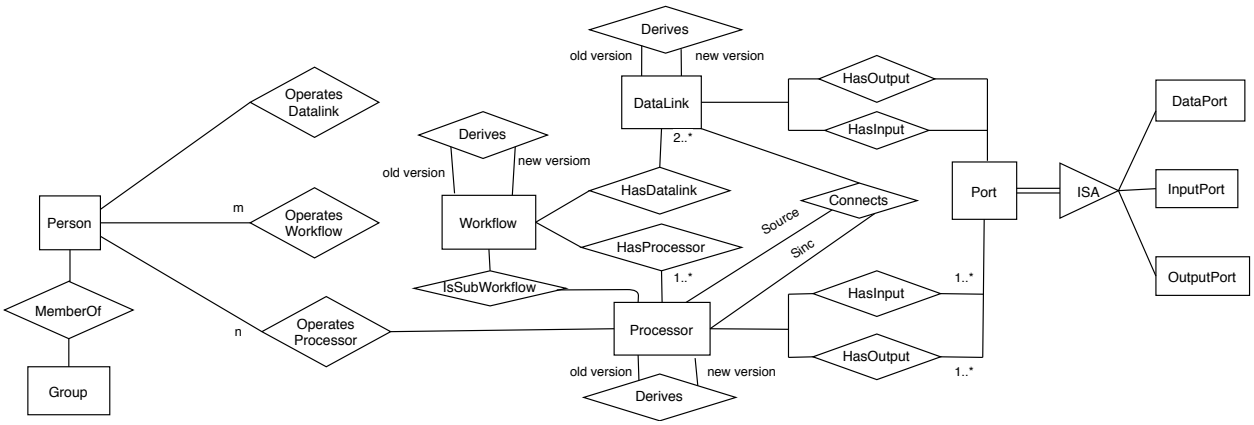

Figure 2: Collaborative Provenance Model (CPM) [68]

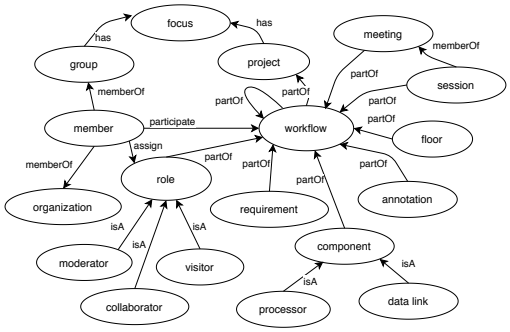

Figure 3: Collaborative workflow composition provenance ontology [70]

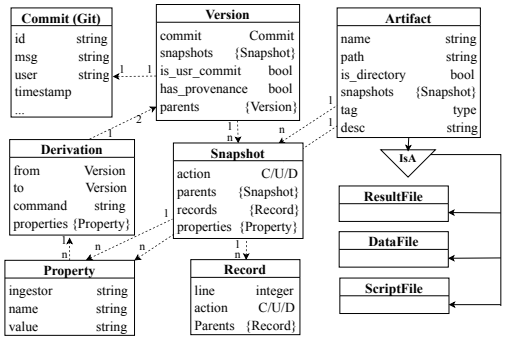

Figure 4: ProvDB Conceptual Data Model [41]

a collaborative workflow composition. The provenance is stored in a provenance repository on the central node of Confucius. Note that the ontology can represent workflows and their components, and roles of people in the collaboration groups. As for Ragan et al.'s classification [51], this model can represent *data* and *interaction* provenance, besides the remaining types through *annotations*.

ProvDB [41] proposes a provenance model with a schema-later approach, providing a base schema that can be extended by arbitrary properties as key-value pairs (Figure 4). Note that these values can be complex, such as a JSON document. The information to the base schema is collected through Git and the built-in ingestors, and additional information can be added through custom ingestors or by user's annotations. When the user runs a command using ProvDB, the system verifies the registered ingestors and executes them. The ingestors can analyze the before- and after-state of the artifacts produced by the command to generate provenance information about the executed command. The model deals with *data* and *interaction* provenance [51] and can deal with all other types of provenance using the ingestors.

Table 1 summarizes how each model supports the collaboration aspects mentioned at the beginning of this section. All the models present limitations when representing some aspects of collaboration. Altintas et al. [1, 2] present a model capable of capturing user collaborations but lack support for the other analyzed items. Confucius [70] and CPM [68] do not adequately treat the heterogeneity of collaboration, not being able to deal with different workflow formats. Confucius also does not deal with workflow evolution. Missier et al. [43] present limitations in dealing with workflow evolution and representing the multilevel hierarchy. ProvDB [41] is the only one providing support for all the analyzed aspects, but it does that making use of extended properties in a key-value schema. Regarding Ragan et al.'s [51] classification, only Confucius and ProvDB can capture all types of provenance, but they do that by using annotations or extended properties. This kind of schema could make things hard and inefficient to query. Another important aspect

Table 1: Summary of the Collaborative Provenance Models

Provenance Model	Provenance Types [51]	Aspects of Collaboration			
		D	H	M	C
Altintas et al. [1, 2]	Data; Interaction	No	No	No	Yes
CPM [68]	Data; Interaction	Yes	Evolution Only	Yes	Yes
Missier et al. [43]	Data; Interaction	Yes	Different schema only	No	Yes
Confucius [38, 61, 67, 70]	All*	Yes	No	Yes	Yes
ProvDB [41]	All*	Yes*	Yes*	Yes*	Yes

*Modeled as extended properties

is that the models just provide a form of storing the information generated in collaborative research and do not necessarily provide a way of collecting them. We also notice that the models supported by a tool [41, 68, 70] can store some provenance on collaboration, but the tool may not fully capture it.

In this section, we show several provenance models that are able to store in part (or in total) collaboration aspects of scientific experiments. However, in order to properly answer our two research questions, we need more insights. In the next section, we discuss how the existing approaches capture and use this information to foster collaboration.

3. COLLABORATION IN SCIENTIFIC RESEARCH

Scientific research is a complex activity per se, and collaboration in this environment becomes a challenging task. To better understand these challenges, we independently analyze the aspects that may influence collaboration in the scientific research scenario. We develop a taxonomy (Figure 5) by examining the 20 approaches we selected, capturing, and categorizing their similarities and differences. We then standardize and enrich the categorization based on other publications [39, 50, 53].

The first branch of the taxonomy is *Experiment Phases*, which is defined in different ways by different authors [6, 39]. In this survey, we use the classification proposed by Mattoso et al. [39], where scientific experiments go through three phases: *composition*, *execution*, and *analysis*. During *composition*, scientists structure and configure the entire experiment, establishing the logical sequence of activities, the type of input data to be provided, and the type of output data. During *execution*, scientists materialize the experiment, define the required input data to run the experiment, trigger its execution (usually carried out by an Experiment Management System), and get the results to be analyzed. During *analysis*, scientists study the gathered data from prior phases [39] aiming at proving or refuting their hypothesis. Each of these experiment phases may involve different forms of collaboration, as discussed in Section 3.1. Provenance plays an important role in each phase, so it is important to keep track of all the user interaction and data transformations on a provenance database.

The second branch of the taxonomy regards the *temporal* aspect of collaboration. This aspect is related to the experience of time and the temporal organization of activities [53]. In a collaborative environment, some tasks need to be synchronized, while others can be done asynchronously. Section 3.2 analyzes if and how existing approaches allow collaborative tasks to occur in real-time or asynchronously.

The third branch is *concurrency control*, which has been extensively studied in the context of databases [52, 15, 23], operating systems [58], and software development [55, 9, 40]. Although the conduction of scientific experiments has its peculiarities, the taxonomy uses ideas that govern version control systems once the problems that may arise when accessing a resource during an experiment resembles the ones that are dealt with by such systems. There are two main concurrency control policies to allow simultaneous work on version control systems: optimistic and pessimistic policy [50]. In pessimistic policies, the artifact that needs to be accessed by several users is restricted to be changed by a single user at a time (i.e., the artifact is locked to a specific user and is only released when the interaction is finished). In optimistic policies, artifacts can be updated in parallel, and users need to merge the changes when conflicts occur. Each of these policies has advantages and disadvantages, and the choice of the most appropriate policy depends on the concurrency frequency, as well as the effort required to merge the artifacts [50]. Section 3.3 discusses how existing approaches deal with concurrency control.

The fourth branch of the taxonomy regards the *sharing* of conceived ideas as well as results and experiments. This allows other researchers to develop new research using these ideas [8]. Although this process is practically mandatory in research, there

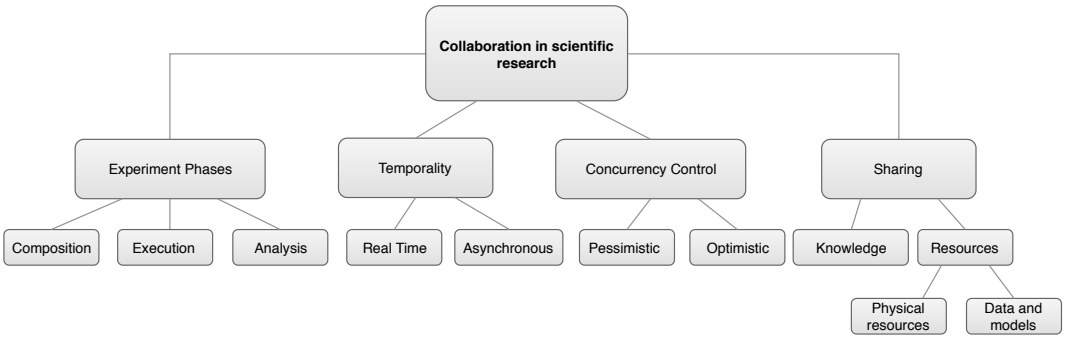

Figure 5: Taxonomy of collaboration in scientific research

is a considerable variation in what is shared, which may facilitate or hinder the research reuse. Some forms of sharing within research would be knowledge sharing, as in publications; data and models sharing, such as sharing a database obtained after some research; and physical resources sharing, such as what happens in the case of institutions sharing a supercomputer. For these different types of sharing (in particular, knowledge, data, or models) to succeed, provenance data is crucial. Without it, the shared information comes out of context and may be useless. Section 3.4 evaluates which of these sharing forms the existing approaches are prepared to deal with, and how this occurs.

Note that all branches of this taxonomy are connected to challenges C1 and C2. They need to be taken into consideration both when collecting provenance (C1) and using this provenance to make collaboration easier (C2). Note also that all branches of the taxonomy are related to data and interaction provenance [51].

Table 2 presents the selected approaches and classifies them according to our taxonomy. This classification considers the aspects addressed in each approach and not the solution maturity of a specific aspect. Thus, two solutions can be equivalently classified, but this does not mean they have the same robustness level. We also evaluated if these tools collect provenance and, when it is possible, classify which type of provenance these tools support. On the next subsections, we detail each of the taxonomy branches and how the surveyed approaches fit them, besides briefly discussing the provenance support of those tools.

3.1 Experiment Phases

Most of the approaches tackle collaboration in the composition phase, while the execution and analysis phases have been receiving less attention.

Composition. This phase has two sub-phases: *conception* and *reuse* [39]. *Conception* aims at producing a high-level representation of the scientific experiment protocol, which is afterward refined and instantiated as a concrete implementation [39] in the form of a workflow or script. *Reuse* consists of retrieving an existing component and adapting it to a new purpose [39].

Some proposals support the *conception* sub-phase [26, 22, 68, 70, 32, 46, 41, 13]. VisTrails [26] is a provenance-aware Workflow Management System that implements support for the collaborative composition of the workflow. Ellkvist et al. [22] and Zhang et al. [68] introduce VisTrails extensions that unleash real-time collaboration on the composition phase of the experiment. Confucius [70] extends Taverna [32] to allow the collaborative composition of workflows by using a client-server architecture that communicates using a service-oriented architecture and XML messages. Mostaeen et al. [46] propose a fine-grained lock scheme that aims to increase efficiency in workflow conception by reducing the waiting time for lock release. ProvDB [41] uses Git to allow the user to collaborate on experiment conception. It also enriches the information collected using ingestors. CoCalc is a virtual workspace for calculations, research, collaboration, and for authoring documents [13], which provides a web portal where scientists can share files with multiple collaborators. This includes Jupyther notebooks, where multiple scientists can simultaneously edit scripts in real-time.

Regarding the *reuse* sub-phase, many of the selected publications focus on the sharing aspect, thus allowing scientists to share a component, a workflow, or a dataset with their peers. That is the case of CAMERA [4], e-ScienceNet [12], myExperiment [28], OpenML [62], Dataverse [35], Collaborative PL-Science [48] and ViroLab [7]. ViroLab [7] provides a way for sharing script components of a workflow. The remaining approaches focus on experiments represented as workflows.

Execution. RASA [42] is the only solution that

Table 2: Aspects of Collaboration in the surveyed Approaches

Approach	Aspects of collaboration				
	Experiment Phase	**Temporality**	**Concurrency Control**	**Sharing**	**Provenance Support**
Confucius [38, 61, 67, 70]	Composition	Asynchronous; Real Time	Pessimistic	Data and models	Data; Interaction
myExperiment [19, 28, 29]	Composition and Analysis	Asynchronous	N/A	Data and models; Knowledge	Yes**
CAMERA [4, 57]	Composition and Analysis	Asynchronous	N/A	Data and models; Knowledge	Yes**
e-ScienceNet [10, 11, 12]	Composition	Asynchronous	N/A	Data and models; Knowledge	No
Collaborative PL-Science [48]	Composition and Analysis	Asynchronous	N/A	Data and models; Knowledge	No
Ellkvist et al. [22]	Composition	Real Time	Optimistic	Data and models	Data
VisTrails [26]	Composition	Asynchronous	Optimistic	N/A	Data
NoCoV [63]	Analysis	Asynchronous; Real Time	N/A	N/A	No
RASA [42]	Execution	Asynchronous	N/A	Physical resources	No
Wood, Wright, and Brodlie [65]	Analysis	Real Time	N/A	N/A	No
ViroLab [7]	Composition	Asynchronous	N/A	Data and models	Yes*
J. Zhang et al [68]	Composition	Real Time	Pessimistic	Data and models	Data; Interaction
Mostaeen et al. [46]	Composition	N/A	Pessimistic	N/A	No
ProvDB [41]	Composition	Asynchronous	Optimistic	Data and models	Data; Interaction
Dataverse [35]	Composition and Analysis	Asynchronous	N/A	Data and models	Yes**
OpenML [62]	Composition and Analysis	Asynchronous	N/A	Data and models	No
CoCalc [13]	Composition and Analysis	Asynchronous; Real Time	Optimistic	Data and models; Knowledge	Data; Interaction
Sumatra [17]	Analysis	Real Time	N/A	Data and models	Data

*No details are provided to correctly classify which provenance types are collected
**Stores data collected by other tools

addresses collaboration in the execution phase of the experiment. RASA is a framework that coordinates the use of scientific instruments, being able to dynamically adapting workflows during the experiment execution according to the needs of the scientists and the equipment.

Analysis. The analysis phase has three sub-phases: *query*, *visualization*, and *discovery* [39]. During *Query*, scientists can relate data and extract information of both the experiment results and provenance data. *Visualization* simplifies the analysis of large volumes of raw data. Data is often projected in graphs or maps to simplify the identification of patterns and the reasoning over the data. During *discovery*, scientists evaluate query results and visual data to draw conclusions about the entire experiment, aiming at checking if the hypothesis is likely to be correct or if it should be refuted. For this, scientists must analyze the experiment as a whole, including all the executions of the experiment (trials) [47].

OpenML [62], CAMERA [4] and myExperiment [28] provide *query* support. They offer a mechanism for sharing not just the workflow components but also other data, such as results and provenance datasets. The myExperiment platform also allows scientists to interact with each other and discuss the shared results. These approaches support the *discovery* sub-phase since they provide a mechanism to analyze and discuss the experiment as a whole. Although not described in the paper [17], Sumatra provides some support to collaboration [56]. It allows different users to share the same provenance database and provides some query features to support the *query* sub-phase.

NoCoV [63] and Wood, Wright, and Brodlie [65] support the *visualization* sub-phase. NoCoV (Notification-service-based Collaborative Visualization) uses a client-server architecture to provide mechanisms for the collaborative visualization of experi-

ment data. The pipeline controller (server) is responsible for synchronizing the clients' visualization, and multiple clients can connect to it simultaneously. The clients could be a pipeline editor (which can update the visualization pipeline) or a parameter control client (which can only adjust visualization parameters). Wood, Wright, and Brodlie [65] propose a collaborative approach on top of IRIS Explorer [24] that allows multiple scientists to interact over a visualization collaboratively.

CoCalc [13] supports the *query, discovery, and visualization* sub-phases. It allows scientists to query the results of the experiment and its history, besides other data. Scientists can also visualize the results using Jupyther notebooks and libraries, such as matplotlib. They can also use chat rooms to discuss the experiment and reason about it.

Dataverse [35] focuses on creating an infrastructure to share datasets related to scientific publications. It provides the data to be used in the *query, discovery, and visualization* sub-phases, although it does not explicitly deal with them.

3.2 Temporality

Starting with the approaches that implement *asynchronous* interactions, CAMERA [4], myExperiment [28], e-ScienceNet [10], Collaborative PL-Science [48], ViroLab [7], Dataverse [35] and OpenML [62] provide solutions focused on the sharing of data and components, where a scientist can publish workflows, components or datasets. These published artifacts become available for other scientists to reuse them *asynchronously*. On VisTrails [26], each version of the workflow is treated as a node in a version tree. Nodes are never modified or deleted (each modification generates a new node in the tree). To collaboratively compose a workflow, scientists can asynchronously work in their local copy of the workflow and synchronize it with another scientist's copy when needed. However, if two scientists modify the same workflow before synchronizing it, this generates multiple disjoint versions, which can be problematic since the changes could be complementary. When this occurs, the scientist should re-implement part of the workflow. ProvDB [41] is a client-server application that uses Git to support version management tasks as well as distributed and decentralized management of individual repositories. Each user makes the necessary modifications to her local repository and, asynchronously, synchronizes them using Git.

We have also identified several proposals that provide *real-time* collaboration. Ellkvist et al. [22] implement a solution based on a client/server architecture, where the server is a MySQL database, and the client is a modified version of VisTrails, that consists of a mechanism to unleash real-time collaboration during workflow composition. The server is used as a shared database to synchronize the versions among the scientists. When one scientist makes a modification, it is saved on the shared database and the other clients are automatically notified to update their local versions. Although implemented in VisTrails, the authors argue that their solution could be implemented in other provenance-aware Workflow Management Systems. Zhang et al. [68] also implement a plugin to VisTrails, which allows any changes made by one scientist to be immediately reflected on all other collaborators' screens. The approach communicates with VisTrails through third-party packages and the VisTrails API. It utilizes Git to provide a new version tree over the existing VisTrails History View. Wood, Wright, and Brodlie [65] present a real-time approach based on a client-server architecture, which allows scientists to visualize an experiment collaboratively. Users can share and alter visualization parameters and visualization pipelines so they can see other users' changes in real-time. Participants may also disconnect single modules from their group to allow periods of independent work on a subset of the pipeline while remaining in contact with the rest of the session. Sumatra [17] provides a way of sharing the provenance database in real-time. The information is shared as soon as it is collected. However, the solution still has several limitations and, in some scenarios, even data loss is possible.

Three solutions work in both real-time and asynchronous scenarios. Confucius [70] provides a solution inspired by a protocol of human communication called Robert's Rules of Order, which is a set of rules created by Henry M. Robert in 1876 to run effective and orderly meetings with maximum fairness to all members [33]. Confucius implements that with a locking strategy that controls which scientist has the right to interact at a given time in a real-time collaboration session. Confucius also maintains a database on the central node that is used for storing provenance of collaboration and workflow evolution, which allows asynchronous collaboration. NoCoV [63] is implemented in a service-oriented architecture that uses notification Web services to synchronize clients and server. When someone alters the visualization pipeline, the pipeline controller notifies other clients, so everyone sees the same visualization in real-time. To transmit information between the pipeline controller and the client, it uses skML [21], an XML-based dataflow

description language. NoCoV uses the stateful Web Services provided by GlobusToolkit 4 (GT4) [59]. Using this stateful feature, the state of the pipeline is persisted and users can retrieve the saved pipeline to continue the work of other users, thus achieving asynchronous collaboration. CoCalc [13] provides a solution based on a web portal where scientists can simultaneously compose scripts in real-time. All changes are immediately synchronized with others. It saves files and data in its cloud infrastructure, so scientists can leave the session and rejoin when needed (allowing asynchronous work).

3.3 Concurrency Control

All approaches providing a mechanism for concurrency control focus on the composition phase of the scientific experiment.

Starting with the approaches that implement the *pessimistic* policy for concurrency control, Confucius' authors [70] treat the concurrency control problem as they would treat it in a face-to-face activity. A central node is needed for the collaboration to occur. A group is registered on this node, and the person responsible for registering the group is automatically assigned as the group moderator. The moderator is responsible for shift control, which is the definition of which group member is allowed to change the workflow at a given time. There is an algorithm for automatically granting and releasing the right to the shift, but the moderator can intervene by taking the right to the shift. Confucius also considers that workflow development can last for long periods in an asynchronous form and, in this scenario, workflow level locking may not be appropriate. Therefore, Confucius blocks smaller building blocks. Thus, several scientists can change the same workflow at the same time. Confucius establishes that the smallest building blocks are tasks and data channels, that in Taverna are called processors and data links, respectively. Confucius introduces the concept of synchronization area "that represents a conceptual area in a shared scientific workflow, which allows only one collaborator to work on it at a given time" [70]. When the user starts to modify a data link, the synchronization area is the data link itself. When the user locks a processor, the synchronization area is the processor and all the fan-out data links of the processor. Zhang et al. [68] also implement a pessimistic collaboration protocol based on Robert's Rules of Order. The protocol is fully described in [34, 69]. Mostaeen et al. [46] analyze the existing locking schemes in terms of concurrency control on the composition of workflows. The approach presents a pessimistic strategy of fine-grain locking in scientific workflows. The lock is done for a single user but at the attribute level, while other approaches use turns or module level locking. The main benefit here is to reduce the waiting time for a lock since smaller portions of the workflow are locked for each modification.

Only four approaches implement the *optimistic* policy for concurrency control. Ellkvist et al. [22] and VisTrails [26] present an optimistic lock approach that creates different branches in the version tree in the case of simultaneous changes. Although VisTrails presents a mechanism for merging, it merges two version trees of different files and not two branches of the same version tree. If the scientists want to keep both of the changes, they will have to use the diff functionality to better understand what has changed and to replay the changes manually. VisTrails also has a functionality called 'analogy' that could help on the process: given two versions of a workflow, VisTrails can automatically detect their differences and apply those differences to another workflow version. Ellkvist et al.'s proposal [22] is built on top of VisTrails, and although it adds support to real-time collaboration, it uses the same concurrency control approach of VisTrails. ProvDB [41] also works on the idea of immutable versions, in which any update will result in a new version. In Cocalc [13], the whole experiment environment is cloud-based. All changes are made directly in the cloud and synchronized with the online scientists' browser – there is no lock.

3.4 Sharing

Most of the approaches providing sharing features allow the sharing of *data and models*. That is the case of e-ScienceNet [12], ViroLab [7], myExperiment [29], CAMERA [4], Dataverse [35], OpenML [62], ProvDB [41], Zhang et al. [68], Ellkvist et al. [22], Confucius [70], Collaborative PL-Science [48], CoCalc [13] and Sumatra [17]. ProvDB [41], Zhang et al. [68], Ellkvist et al. [22], and Confucius [70] work with a centralized database for the experiment, which stores the provenance collected from the collaborative experiment and makes this information available to the involved scientists. ViroLab [7] addresses the issue of sharing code blocks for reuse. The approach also mentions the persistence and sharing of provenance but does not provide details on what kind of provenance information is stored and shared. Sumatra [17] provides a way of sharing a provenance database between multiple scientists.

Roure, Globe, and Stevens [19] argue that one of the barriers of workflow reuse is on how the knowl-

edge about the workflow could be transmitted to potential users. That challenge can be minimized by the distribution of other documentation data in addition to the workflow definition. Most of the approaches try to increase collaboration by adding the possibility of sharing *knowledge*. That is the case of e-ScienceNet [12], myExperiment [29], CAMERA [4], Dataverse [35], CoCalc [13], and Collaborative PL-Science [48]. Pereira et al. [48] propose the Collaborative PL-Science, an extension of PL-Science [14]. It aims to facilitate the reuse of components in the construction of scientific workflows, thus combining models and knowledge sharing. The idea is that adding information that helps to understand published artifacts facilitates reuse. The approach uses ontologies to enrich the information of shared objects. CoCalc [13] allows the sharing of a great variety of files, including scripts in multiple programming languages. It also allows the sharing of documentation that can help scientists to better understand what has been made on the experiment and help them to better use the shared data and scripts. e-ScienceNet [12] is another approach that allows both the sharing of data and models and also knowledge. It differs from other approaches because it presents a peer-to-peer solution for sharing the experiment results and models without the dependency of a central server.

Some publications explore the creation of portals for sharing data and reusable components in research, where it is common to share scientific workflows. Goble and Roure [29] propose myExperiment, a social network for scientists focused on workflow-related issues. It allows the sharing of the workflow itself as well as other metadata, such as provenance logs, besides enabling researchers to interact using the tool, commenting, and discussing the shared resources. CAMERA [4] also focuses on the sharing of scientific workflows and provenance logs. The tool works exclusively with Kepler [3] workflows and allows the execution of the experiments within the portal. OpenML [62] is focused on the machine learning community and provides a portal to share datasets, algorithm implementations, and workflows. It also presents a Web API, which allows users to interact with the portal in a programmatic form, and ways of sharing scientific tasks and receiving other scientists' collaboration. Dataverse [35] provides a Web infrastructure to share datasets related to scientific publications. The main idea is that sharing the datasets may increase the reproducibility of experiments, and, as a counterpart to the authors, it may increase the number of citations of the related publications [35].

RASA [42] is the only approach that focuses on sharing physical resources. The approach provides a framework for coordinating the use of scientific instruments. The idea is to provide a mechanism to dynamically modify workflows depending on the needs of the requester scientist and the particularities of the equipment, and also the knowledge of the equipment operator.

3.5 Provenance Support

As seen in Table 2, many of the tools do not collect provenance. Although ViroLab [7] provides some provenance support, it does not give details on what is stored. Dataverse [35], CAMERA [4], and myExperiment [19] provide support for storing and sharing provenance data collected by other tools. CoCalc [13] collects interaction provenance through the log of the activities executed by scientists, but this unstructured information is hard to query. VisTrails [26], Ellkvist et al. [22], and Sumatra [17] can capture data provenance from multiple users in their local stations and consolidate them in a single database, but those databases do not properly represent collaboration aspects of the research covered by Section 2, thus collaboration provenance is not included. Zhang et al. [68], Confucius [38, 61, 67, 70], and ProvDB [41] provide data and interaction provenance support, and use the collaboration-aware provenance models described in Section 2. The models proposed by Confucius and ProvDB need extended properties to represent some collaboration aspects, but the tools proposed by those papers are not able to capture these properties. Thus, there is a difference between the provenance types represented by the models and those supported by the tools.

4. DISCUSSION AND OPPORTUNITIES

Figure 6 shows a timeline that helps understand how research has progressed in this field. Some of the publications are highly related and represent the evolution of the same research. In such cases, we treat them in a consolidated manner, thus linking these publications in the figure and handling them as a single approach. This topic has received much attention in recent years, but there are still some gaps to be further explored. In this section, we classify the selected approaches, answer the research questions introduced in Section 1, discuss the gaps that still exist, and present opportunities derived from those gaps.

R1: How do existing tools store and collect provenance in a collaborative experiment?

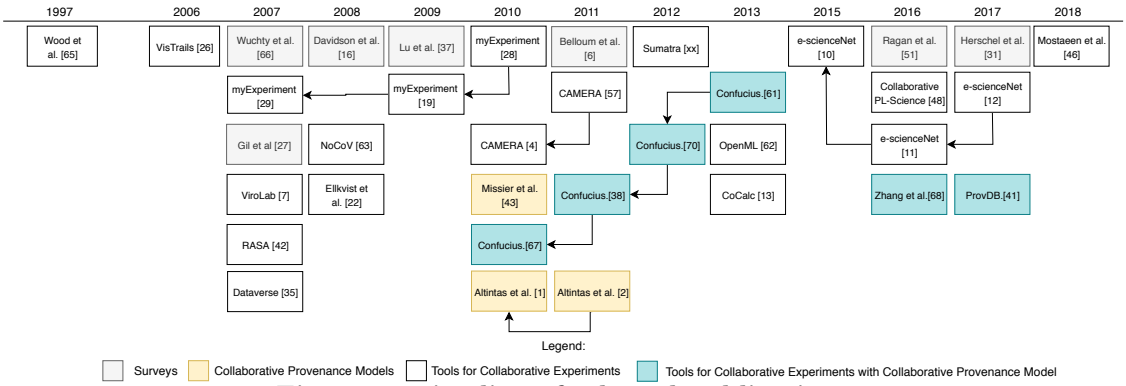

Figure 6: Timeline of selected publications

To answer this question, we analyzed the available models for storing provenance in collaborative environments. Although significant progress has been made with those models, all of them present limitations (they do not deal with different workflow formats, or do not deal with workflow evolution). Models that can represent all the aspects we analyzed do so by using extended properties, which makes them difficult to query.

Regarding the available tools and how they collect provenance: Some tools (Dataverse [35], CAMERA [4], and myExperiment [19]) just provide storage for provenance, but do not collect it. Other tools (VisTrails [26] , Ellkvist et al. [22], and Sumatra [17]) provide a way of consolidating the provenance collected from different users but lack support for other collaboration aspects. Finally, a few tools (Zhang et al. [68], Confucius [38, 61, 67, 70], and ProvDB [41]) use collaborative aware provenance models but still present some limitations.

R2: How do existing tools use provenance to make collaboration easier in scientific experiments? We conclude that the surveyed approaches fail to use the collected provenance to support the collaboration. Although Confucius [70], Zhang et al. [68], and ProvDB [41] are capable of collecting provenance of the collaboration process, they do not propose forms of using that valuable data to increase the efficiency and awareness of the process.

As illustrated in Table 2, most of the approaches support the composition phase of the experiment life cycle (especially the conception sub-phase). However, they are mostly based on Workflow Management Systems and ignore the fact that many scientists use scripts in their experiments [49]. The only approaches that support experiments represented as scripts are ViroLab [7], Sumatra [17], CoCalc [13], and ProvDB [41]. However, ViroLab only addresses the reuse sub-phase of the experiment composition. Sumatra fully delegates the script composition to Git and presents several limitations for the shared provenance storage, such as a possible data loss depending on the network connection. Despite being quite complete, CoCalc [13] demands the scientist to be online in order to work, and that she works on the browser, which can be a tough change in the workspace, tools, and IDEs that the scientist is used to. It is possible to run applications from the CoCalc portal, but this is not the same as running them from the scientist's machine. It also presents several limitations on free accounts. Another point worth mentioning is that it does not properly capture the provenance of the experiment. It presents features like "time travel" and "log" that let users see the history of the files and activity on the project, but it is very high level and may not be enough to guarantee the reproducibility of the experiment, for example. ProvDB uses Git to handle version management and a provenance ingestor framework to capture other provenance data, but it is highly specialized in data science problems and is not well prepared for a general-purpose experiment.

Although versioning tools handle several collaborative needs of script building, they are software development tools that do not address specific problems in scientific research. These tools will not provide provenance capture and analysis support by default. Provenance is not just related to the obtained results but also the input data, intermediate results, etc. Trying to deal with this complexity without the proper tooling support could take much effort from the scientists and steal the energy that should be spent on research. Although ProvDB considers these challenges, it depends on the scientist being able to access an external tool (Git), a specific OS (UNIX), and demands the creation of ingestors to capture some provenance aspects. ProvDB is also focused on a specialized type of experiment (data science analysis), and does not address awareness

during collaboration. Thus, ***we must investigate and design provenance-aware tools that can handle composition, execution, and analyses of generic script-based experiments collaboratively, increasing the awareness of users during the process at the same time***.

The execution phase also lacks support. We could find only one approach that supports collaboration in this phase of the experiment life cycle. RASA [42] supports the execution phase by controlling access to physical resources such as equipment. Providing provenance-aware support of the execution phase is crucial in collaborative experiments, since without it, important aspects of the collaboration may be lost. In fact, for reproducibility purposes, it is crucial to know which user executed each part of the experiment, where and under which conditions. Thus, the ***support for the collaborative execution of scientific experiments needs more investigation***.

Some approaches support the analysis phase of experiments. Most of them allow scientists to comment on the experiment structure or results. Some approaches [7, 26, 41, 70] provide provenance gathering of the collaborative experiment that could help the analysis of the experiment. However, they do not provide a clear way to collaborate throughout the analysis, so they were classified without this phase of the life cycle in Table 2.

Temporality is well explored, with several approaches supporting asynchronous or real-time interactions. However, some features could be improved. When conducting an experiment in groups, it is important to know what happened in the experiment while scientists were offline, who did what, and in which part of the experiment (interaction provenance). It is also important to know if there is anyone online and in which part of the experiment they are working at. Although some tools let the users query for some of that information, it would be desirable that such information would be automatically shown to users, depending on the context of the experiment. Thus, ***an interesting issue to examine will be ways of increasing the awareness of the scientists about the actions of their collaborators***.

As for concurrency control, most of the approaches use a pessimistic locking scheme. Pessimistic locking may work well in real-time scenarios, but it can be quite troublesome for asynchronous collaboration. VisTrails [26] and Ellkvist et al. [22] are the only solutions that work with an optimistic locking scheme, but they do not implement a merging mechanism capable of merging two workflow branches. Although VisTrails diff and analogy functionalities could help to merge two branches, they impose some additional steps for such a task and lack some basic merge functionalities like conflict resolution. Thus,***we need tools that work with optimistic locking and provide complete merge support in the composition of workflows***.

Also, in a collaborative environment, some collaboration tasks may perform better if treated with a pessimistic locking policy while others will benefit from an optimistic approach [50]. In experiments with files that are difficult to merge, scientists could opt to work with a pessimistic policy, while in others they may prefer to work with an optimistic one. Existing tools only implement one of the policies, so if scientists want to use this tool, they are forced to use the implemented policy. Scientists must have the flexibility needed to interact in a way that is more appropriate to the use case in hand. Thus, ***tools that allow scientists to choose the more appropriate lock policy are needed***.

Sharing is well covered in the literature with a wide range of available solutions. Solutions address centralized sharing as well as peer-to-peer sharing, besides providing mechanisms for commenting and enriching the shared artifacts, making them easier to use. We believe that, in this aspect, there is no clear gap in the available tools.

We end up finding that none of the available tools are capable of using provenance to make collaboration easier in scientific experiments (related to R2). So, ***there is a need to investigate how to use the captured provenance to make collaboration easier in scientific research***.

5. CONCLUSION

Scientific research is frequently collaborative and also conducted in silico. Although this is very positive for science, it brings several challenges. To better understand the challenges and evaluate the literature on the subject, this article presents a survey on collaboration in in silico scientific research. In this survey, we map the available tools and the state-of-art of research on collaborative experiments conducted in silico. We propose a taxonomy and use it to classify the existing tools and discuss opportunities based on the gaps we identified. We believe that a more systematic review process could find new articles and enrich the obtained results. However, we believe we cover a large part of the publications on the topic, and our findings at this stage can be useful and generate insights to researchers interested in this topic.

6. REFERENCES

[1] I. Altintas, M. K. Anand, D. Crawl, S. Bowers, A. Belloum, P. Missier, B. Ludäscher, C. A. Goble, and P. M. A. Sloot. Understanding collaborative studies through interoperable workflow provenance. In D. L. McGuinness, J. R. Michaelis, and L. Moreau, editors, *Provenance and Annotation of Data and Processes*, pages 42–58. Springer Berlin Heidelberg, 2010.

[2] I. Altintas, M. K. Anand, T. N. Vuong, S. Bowers, B. Ludäscher, and P. M. A. Sloot. A data model for analyzing user collaborations in workflow-driven escience. *International Journal of Computers and Their Applications*, 18:160–179, 2011.

[3] I. Altintas, C. Berkley, E. Jaeger, M. Jones, B. Ludascher, and S. Mock. Kepler: an extensible system for design and execution of scientific workflows. In *Scientific and Statistical Database Management*, pages 423–424, 2004.

[4] I. Altintas, A. W. Lin, J. Chen, C. Churas, M. Gujral, S. Sun, W. Li, R. Manansala, M. Sedova, J. S. Grethe, and M. Ellisman. Camera 2.0: A data-centric metagenomics community infrastructure driven by scientific workflows. In *World Congress on Services*, pages 352–359, 2010.

[5] M. K. Anand, S. Bowers, T. McPhillips, and B. Ludäscher. Exploring scientific workflow provenance using hybrid queries over nested data and lineage graphs. In M. Winslett, editor, *Scientific and Statistical Database Management*, Lecture Notes in Computer Science, pages 237–254. Springer Berlin Heidelberg, 2009.

[6] A. Belloum, M. A. Inda, D. Vasunin, V. Korkhov, Z. Zhao, H. Rauwerda, T. M. Breit, M. Bubak, and L. O. Hertzberger. Collaborative e-science experiments and scientific workflows. *IEEE Internet Computing*, 15(4):39–47, July 2011.

[7] M. Bubak, T. Gubala, M. Kasztelnik, M. Malawski, P. Nowakowski, and P. Sloot. Collaborative virtual laboratory for e-health. In *Expanding the Knowledge Economy: Issues, Applications, Case Studies, eChallenges*, pages 537–544, 2007.

[8] R. Caldwell and D. Lindberg. Participants in science behave scientifically. *Understanding Science.*, 2018. Available at `https://undsci.berkeley.edu/article/0_0_0/whatisscience_09`.

[9] S. Chacon and J. Long. Git. `https://git-scm.com/`. Accessed: 2018-06-09.

[10] T. Classe, R. Braga, F. Campos, and J. M. N. David. A semantic peer to peer network to support e-science. In *IEEE International Conference on e-Science*, pages 503–512, 2015.

[11] T. Classe, R. Braga, J. M. N. David, F. Campos, M. A. Araújo, and V. Ströele. A collaborative approach to support e-science activities. In *IEEE International Conference on Computer Supported Cooperative Work in Design*, pages 20–25. IEEE, 2016.

[12] T. Classe, R. Braga, J. M. N. David, F. Campos, and W. Arbex. A distributed infrastructure to support scientific experiments. *Journal of Grid Computing*, 15(4):475–500, 2017.

[13] Cocalc user manual documentation. `https://doc.cocalc.com/contents.html`, 2013. Accessed: 2019-12-05.

[14] G. C. B. Costa, R. Braga, J. M. N. David, and F. Campos. A scientific software product line for the bioinformatics domain. *Journal of Biomedical Informatics*, 56:239–264, 2015.

[15] C. J. Date. *An introduction to database systems*. Pearson/Addison Wesley, Boston, 2004.

[16] S. B. Davidson and J. Freire. Provenance and scientific workflows: Challenges and opportunities. In *ACM Special Interest Group on Management of Data*, pages 1345–1350. ACM, 2008.

[17] A. Davison. Automated capture of experiment context for easier reproducibility in computational research. *Computing in Science & Engineering*, 14(4):48–56, 2012.

[18] D. De Oliveira, E. Ogasawara, F. Baião, and M. Mattoso. Scicumulus: A lightweight cloud middleware to explore many task computing paradigm in scientific workflows. In *International Conference on Cloud Computing*, pages 378–385, Washington, DC, USA, 2010.

[19] D. De Roure, C. Goble, and R. Stevens. The design and realisation of the myexperiment virtual research environment for social sharing of workflows. *Future Generation Computer Systems*, 25(5):561–567, 2009.

[20] E. Deelman, G. Singh, M.-H. Su, J. Blythe, Y. Gil, C. Kesselman, G. Mehta, K. Vahi, G. B. Berriman, J. Good, A. Laity, J. C. Jacob, and D. S. Katz. Pegasus: a framework for mapping complex scientific workflows onto

distributed systems. *Scientific Programming Journal*, 13(3):219–237, 2005.
[21] D. A. Duce and M. S. Sagar. skML a markup language for distributed collaborative visualization. In *Theory and Practice of Computer Graphics*, pages 171–178, 2005.
[22] T. Ellkvist, D. Koop, E. W. Anderson, J. Freire, and C. Silva. Using provenance to support real-time collaborative design of workflows. In *International Workshop on Provenance and Annotation (IPAW)*, pages 266–279. Springer, 2008.
[23] R. Elmasri and S. Navathe. *Fundamentals of database systems*. Addison-Wesley, 6 edition, Apr. 2010.
[24] D. Foulser. IRIS Explorer: a framework for investigation. *SIGGRAPH Computer Graphics*, 29(2):13–16, 1995.
[25] J. Freire, D. Koop, E. Santos, and C. T. Silva. Provenance for computational tasks: A survey. *Computing in Science & Engineering*, 10(3):11–21, 2008.
[26] J. Freire, C. T. Silva, S. P. Callahan, E. Santos, C. E. Scheidegger, and H. T. Vo. Managing rapidly-evolving scientific workflows. In L. Moreau and I. Foster, editors, *Provenance and Annotation of Data*, Lecture Notes in Computer Science, pages 10–18. Springer Berlin Heidelberg, 2006.
[27] Y. Gil, E. Deelman, M. Ellisman, T. Fahringer, G. Fox, D. Gannon, C. Goble, M. Livny, L. Moreau, and J. Myers. Examining the challenges of scientific workflows. *Computer*, 40(12):24–32, 2007.
[28] C. A. Goble, J. Bhagat, S. Aleksejevs, D. Cruickshank, D. Michaelides, D. Newman, M. Borkum, S. Bechhofer, M. Roos, P. Li, and D. De Roure. myexperiment: a repository and social network for the sharing of bioinformatics workflows. *Nucleic Acids Research*, 38(Web Server Issue):677–682, 2010.
[29] C. A. Goble and D. C. D. Roure. myexperiment: Social networking for workflow-using e-scientists. In *Workshop on Workflows in Support of Large-Scale Science*, pages 1–2. ACM, 2007.
[30] L. A. Goodman. Snowball sampling. *The Annals of Mathematical Statistics*, 32(1):148–170, 1961.
[31] M. Herschel, R. Diestelkämper, and H. B. Lahmar. A survey on provenance: What for? what form? what from? *The VLDB Journal*, 26(6):881–906, 2017.
[32] D. Hull, K. Wolstencroft, R. Stevens, C. Goble, M. R. Pocock, P. Li, and T. Oinn. Taverna: a tool for building and running workflows of services. *Nucleic Acids Research*, 34(2):729–732, 2006.
[33] H. M. R. III, D. H. Honemann, T. J. Balch, D. E. Seabold, and S. Gerber. *Robert's rules of order newly revised*. PublicAffairs, 11 edition, 2011.
[34] Jia Zhang, C. Chang, and Jen-Yao Chung. Mediating electronic meetings. In *International Computer Software and Applications Conference*, pages 216–221, 2003.
[35] G. King. An introduction to the dataverse network as an infrastructure for data sharing, 2007.
[36] B. Lerner and E. Boose. Rdatatracker: collecting provenance in an interactive scripting environment. In *USENIX Workshop on the Theory and Practice of Provenance (TaPP)*, 2014.
[37] S. Lu and J. Zhang. Collaborative scientific workflows. In *IEEE International Conference on Web Services*, pages 527–534. IEEE, 2009.
[38] S. Lu and J. Zhang. Collaborative scientific workflows supporting collaborative science. *International Journal of Business Process Integration and Management*, page 185, 2011.
[39] M. Mattoso, C. Werner, G. H. Travassos, V. Braganholo, E. Ogasawara, D. Oliveira, S. Cruz, W. Martinho, and L. Murta. Towards supporting the life cycle of large scale scientific experiments. *International Journal of Business Process Integration and Management*, 5(1):79–92, 2010.
[40] Mercurial scm. `https://www.mercurial-scm.org/`. Accessed: 2019-04-23.
[41] H. Miao, A. Chavan, and A. Deshpande. Provdb: Lifecycle management of collaborative analysis workflows. In *Workshop on Human-In-the-Loop Data Analytics (HILDA)*, pages 7:1–7:6, New York, NY, USA, 2017. ACM.
[42] T. Miller, P. McBurney, J. McGinnis, and K. Stathis. First-class protocols for agent-based coordination of scientific instruments. In *IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises*, pages 41–46, 2007.
[43] P. Missier, B. Ludascher, S. Bowers, S. Dey, A. Sarkar, B. Shrestha, I. Altintas, M. Anand, and C. Goble. Linking multiple workflow provenance traces for interoperable

collaborative science. In *Workshop on Workflows in Support of Large-Scale Science*, pages 1–8, 2010.

[44] L. Moreau, B. Clifford, J. Freire, J. Futrelle, Y. Gil, P. Groth, N. Kwasnikowska, S. Miles, P. Missier, J. Myers, B. Plale, Y. Simmhan, E. Stephan, and J. V. den Bussche. The open provenance model core specification (v1.1). *Future Generation Computer Systems*, 27(6):743–756, 2011.

[45] L. Moreau, P. Missier, K. Belhajjame, R. B'Far, J. Cheney, S. Coppens, S. Cresswell, Y. Gil, P. Groth, G. Klyne, T. Lebo, J. McCusker, S. Miles, J. Myers, S. Sahoo, and C. Tilmes. PROV-DM: The PROV data model. W3C Recommendation. *W3C Recommendation*, 2013. Available at `http://www.w3.org/TR/2013/REC-prov-dm-20130430/`.

[46] G. Mostaeen, B. Roy, C. K. Roy, and K. A. Schneider. Fine-grained attribute level locking scheme for collaborative scientific workflow development. In *IEEE International Conference on Services Computing*, pages 273–277, 2018.

[47] L. Murta, V. Braganholo, F. Chirigati, D. Koop, and J. Freire. noworkflow: Capturing and analyzing provenance of scripts. In *International Workshop on Provenance Annotation (IPAW)*, pages 1–12, 2014.

[48] A. F. Pereira, J. M. N. David, R. Braga, and F. Campos. An architecture to enhance collaboration in scientific software product line. In *International Conference on System Sciences*, pages 338–347. IEEE, 2016.

[49] J. F. Pimentel, J. Freire, L. Murta, and V. Braganholo. A survey on collecting, managing, and analyzing provenance from scripts. *ACM Computing Surveys*, 52(3):47:1–47:38, 2019.

[50] J. Prudêncio, L. Murta, C. Werner, and R. Cepêda. To lock, or not to lock: That is the question. *Journal of Systems and Software*, 85(2):277–289, 2012.

[51] E. D. Ragan, A. Endert, J. Sanyal, and J. Chen. Characterizing provenance in visualization and data analysis: an organizational framework of provenance types and purposes. *IEEE Transactions on Visualization and Computer Graphics*, 22(1):31–40, 2016.

[52] R. Ramakrishnan and J. Gehrke. *Database management systems*. McGraw-Hill, New York, third edition edition, 2003.

[53] M. C. Reddy, P. Dourish, and W. Pratt. Temporality in medical work: Time also matters. *Computer Supported Cooperative Work*, 15(1):29–53, 2006.

[54] D. H. Sonnenwald. Scientific collaboration. *Annual review of information science and technology*, 41(1):643–681, 2007.

[55] Apache subversion. `https://subversion.apache.org/`. Accessed: 2019-04-23.

[56] Sumatra 0.7.0 documentation. `https://pythonhosted.org/Sumatra/record_stores.html`. Accessed: 2019-12-03.

[57] S. Sun, J. Chen, W. Li, I. Altintas, A. Lin, S. Peltier, K. Stocks, E. E. Allen, M. Ellisman, J. Grethe, and J. Wooley. Community cyberinfrastructure for advanced microbial ecology research and analysis: the CAMERA resource. *Nucleic Acids Research*, 39:D546–551, 2011.

[58] A. S. Tanenbaum. *Modern operating systems*. Prentice Hall, Upper Saddle River, N.J, 3 edition edition, Dec. 2007.

[59] Gt4 globus toolkit web site. `http://toolkit.globus.org/toolkit/`. Accessed: 2019-04-23.

[60] G. H. Travassos and M. O. Barros. Contributions of in virtuo and in silico experiments for the future of empirical studies in software engineering. In *Workshop on Empirical Software Engineering the Future of Empirical Studies in Software Engineering*, pages 117–130, 2003.

[61] S. Vali and S. Sreerama. Multi-user tool for scientific work flow composition. *International Journal of Computer Trends & Technology*, 4, 2013.

[62] J. N. Van Rijn, B. Bischl, L. Torgo, B. Gao, V. Umaashankar, S. Fischer, P. Winter, B. Wiswedel, M. R. Berthold, and J. Vanschoren. Openml: A collaborative science platform. In *Joint european conference on machine learning and knowledge discovery in databases*, pages 645–649. Springer, 2013.

[63] H. Wang, K. W. Brodlie, J. W. Handley, and J. D. Wood. Service-oriented approach to collaborative visualization. *Concurrency and Computation: Practice and Experience*, 20(11):1289–1301, 2008.

[64] M. Wilde, I. Foster, K. Iskra, P. Beckman, Z. Zhang, A. Espinosa, M. Hategan, B. Clifford, and I. Raicu. Parallel scripting for applications at the petascale and beyond.

Computer, 42(11):50–60, 2009.

[65] J. Wood, H. Wright, and K. Brodlie. Collaborative visualization. In *Conference on Visualization*, pages 253–259. IEEE Computer Society Press, 1997.

[66] S. Wuchty, B. F. Jones, and B. Uzzi. The increasing dominance of teams in production of knowledge. *Science*, 316(5827):1036–1039, 2007.

[67] J. Zhang. Co-taverna: A tool supporting collaborative scientific workflows. In *IEEE International Conference on Services Computing*, pages 41–48, 2010.

[68] J. Zhang, Q. Bao, X. Duan, S. Lu, L. Xue, R. Shi, and P. Tang. Collaborative scientific workflow composition as a service: An infrastructure supporting collaborative data analytics workflow design and management. In *IEEE International Conference on Collaboration and Internet Computing*, pages 219–228, 2016.

[69] J. Zhang, C. K. Chang, and J. Voas. A uniform meta-model for mediating formal electronic conferences. In *International Computer Software and Applications Conference*, pages 376–381. IEEE, 2004.

[70] J. Zhang, D. Kuc, and S. Lu. Confucius: A tool supporting collaborative scientific workflow composition. *IEEE Transactions on Services Computing*, 7(1), 2012.