


Agile Research Data Management with Open Source: LinkAhead

Daniel Hornung  ¹

Florian Spreckelsen  ¹

Thomas Weiß  ¹


1. IndiScale GmbH, Göttingen.



Date Received:

2021-08-01

Licenses:

This article is licensed under: 

Keywords:

Data Management, Research Data Management, Agile Data Management, Software Tools, FAIR Data, Good Scientific Practice

Data availability:

Software availability:

Software can be found here: <https://gitlab.com/linkahead> and at DOI:10.5281/zenodo.7752417

Abstract.

Research data management (RDM) in academic scientific environments increasingly enters the focus as an important part of good scientific practice and as a topic with big potentials for saving time and money. Nevertheless, there is a shortage of appropriate tools, which fulfill the specific requirements in scientific research. We identified where the requirements in science deviate from other fields and proposed a list of requirements which RDM software should answer to become a viable option.

We analyzed a number of currently available technologies and tool categories for matching these requirements and identified areas where no tools can satisfy researchers' needs. Finally we assessed the open-source RDMS (research data management system) LinkAhead for compatibility with the proposed features and found that it fulfills the requirements in the area of *semantic, flexible data handling* in which other tools show weaknesses.

1 Introduction

2 Research units, from small research groups at universities to large research and development
3 departments are increasingly confronted with the challenge to manage large amounts of data, data
4 of high complexity[1], [2] and changing data structures[3], [4]. The necessary tasks for research
5 data management include storage, findability and long-term accessibility for new generations of
6 researchers and for new research questions[4]–[6].

7 In spite of the advantages of implementing data management solutions[7], we found a lack of
8 standard methods or even standard software so far for research data management, especially in
9 the context of quickly evolving methods and research targets. We hypothesize that the reason for
10 this deficit is that scientific research poses unique challenges for data management, since it is
11 characterized by constant innovation, short lived research questions, trial-and-error approaches,
12 and the continuous integration of new insights.

13 We propose *agile research data management* as a promising approach to meet the special
14 requirements of scientific research and to fully leverage the benefits of increased research
15 digitalization, automated data acquisition methods and storage capabilities.

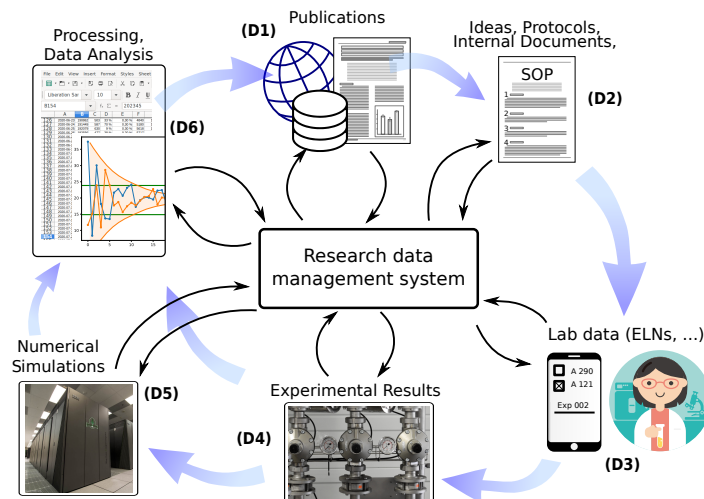


Figure 1: Schematic illustration of the scientific data lifecycle. Data can be obtained from every step, and in most cases the relationship between data entities is just as relevant as the raw data. Blue thick arrows denote the direction in which information flows in normal research. Thin black arrows indicate data flow to and from a research data management system. While this example focuses on experimental and laboratory centered disciplines, comparative lifecycles also exist for theoretical sciences and most fields in the humanities.

16 For this article, we identified the specific challenges for research data management (RDM) and
 17 defined eleven requirements which suitable RDM software should have to (a) fulfill the practical
 18 needs and (b) be accepted by the potential users. We then matched existing tools against these
 19 requirements and found areas where the tools show substantial need for improvement.
 20 Finally we present the LinkAhead[8], [9] toolkit as a viable approach to satisfy all the proposed
 21 requirements.

22 2 Challenges for research data management

23 2.1 The scientific data lifecycle: the need for proper tooling

24 Data which accrues in scientific research is more than just experimental readings, field notes
 25 or interview recordings. In order to fully represent the research journey and eventually enable
 26 reproducible science, the data from every research step may become relevant. We identify
 27 the challenges to make this data usable in a way that leads to reproducible, and time-efficient,
 28 research.

29 Figure 1 shows a schematic of different research steps during the research lifecycle, during
 30 which important data is generated. For full reproducibility, it is not sufficient however to simply
 31 store any data that one acquires, but also to represent the semantic connections and make these
 32 connections searchable.

33 In more detail, the most relevant sources and targets for data in scientific research are (numbered
 34 from (D1) to (D6)):

35 **Prior publications (D1)** An important part of good scientific practice (GSP) is to credit the

36 influence of prior work, written by the scientists themselves or third parties. Linking one's
37 own work to previous publications — articles or published data from repositories — and
38 making these connections public helps to assess reproducibility and may lead to fruitful
39 data-reuse in unforeseen contexts.[10] An RDMS should be able to trace back each data
40 item to previous scientific publications on which it is based.

41 **Ideas and SOPs (D2)** The data here consists mostly of text documents which describe thoughts,
42 hypotheses and planned standard operating procedures (SOPs). These documents fill the
43 gap between previous work and the next round of data acquisitions, they often also work
44 as a blueprint for the data acquisition phase.[11] A scientist may consult their RDMS
45 to answer questions like “Which SOP was used when experiment X was carried out to
46 generate data file Y?”.

47 **Lab data (D3)** Environmental data, device settings, used SOPs and ingredients and other inci-
48 dental data typically accrues during the course of experiments and was traditionally stored
49 in paper laboratory notebooks. Currently, a lot of laboratories switch to electronic lab
50 notebooks (ELNs) for the same purpose. While this data is often seen as second-class
51 “metadata”, we hold that since often conclusions can be drawn from it, it deserves the
52 same handling as final instrument readings.[12]–[14]

53 During work in the lab, software must be as unintrusive as possible, with efficient user
54 interfaces.

55 **Experimental results (D4)** These are what is often considered the *main* data. For meaningful
56 analysis, data from experimental results mostly must be enriched with additional data from
57 experimental or device settings or from processed samples, to filter for special conditions,
58 to compare settings or to verify that values are compatible with standard literature.[15]

59 **Numerical simulations (D5)** Similarly to experimental results, data obtained from numerical
60 procedures can not be interpreted without knowledge about used software and parameters,
61 possibly hardware conditions and input from laboratories or third-party data sources.[16]
62 Since bit-for-bit reproducibility is possible in theory, all relevant settings should be stored
63 unchanged.

64 **Data analysis (D6)** When analyzing data from previous steps, storing not only the used pro-
65 grams, scripts, and their parameters, but also the semantic connections enables later
66 researchers to reconstruct which method was used, which assumptions were made and
67 under which conditions the input data was gathered.[17], [18]

68 **Next publication (D1)** Formally the end of the lifecycle, but of course also the beginning of
69 many new ones, a publication contains a number of statements which are supported by
70 data from previous steps. A comprehensive RDMS could quickly answer a question like
71 “In figure X, which methods were used to analyze the data, which devices and software
72 were used to acquire the raw data, and which assumptions were made when planning the
73 experimental setting?”

74 This list focuses on experimental and laboratory centered disciplines like engineering or natural
75 sciences, but of course in the humanities and theoretical sciences, there are equivalent steps
76 which are equally important to preserve and link to each other.

77 2.2 Specifics of scientific research data management

78 There are some needs for data management which are specific to or more pronounced in scientific
79 research, which we will label by (S1) through (S5):

80 **Interoperability** Scientists tend to work with their own custom-written software[19]–[21],
81 which often requires files with data to be directly accessible to the OS via a file system
82 **(S1)**, remote or locally. Also programmatic access (query, retrieve, update) to data via
83 network APIs **(S2)** is a necessity for many scientific data uses.

84 **Agility** Traditional DMS require users to define a data model and stick to it[22]. All data to be
85 entered has to conform to the data model as it was defined. Research however is defined
86 by having undefined outcomes, the research questions, experimental setup or analysis
87 methods change more often than not over the course of one investigation.[23] We therefore
88 identify **(S3)** as the special need for flexibility regarding the data model.

89 **Learning curve** Scientific research is founded upon the contribution of many participants, with
90 different qualifications, varying research foci and high fluctuations. As a consequence, a
91 steep learning curve for using an RDMS would be detrimental to its adoption **(S4)**.

92 **Early usefulness** Systems which only store data, but do not provide short-term advantages, have
93 high acceptance barriers. Especially in academic research, junior scientists with short-term
94 contracts have little incentive to invest time and money in systems which only may pay
95 out on longer timescales.[22] Therefore, RDMS should offer some tangible advantages on
96 the short run **(S5)**.

97 3 Requirements for a scientific RDMS

98 Based upon the challenges from the previous section, we propose a set of requirements for an
99 RDMS to be a useful tool for scientific research.

100 3.1 General requirements

101 **(R1) Semantic linkage** In order to retain the semantic context in which data is embedded, it
102 must be possible in the RDMS to link data sets with each other in a meaningful way, i.e.,
103 the links must bear some meaning. The default linking possibilities and properties of the
104 data types in the RDMS form the *data model*.

105 **(R2) Flexible data model** Researchers require an RDMS for structured storage of data, where
106 the data model can be changed on the fly, without the need to migrate or discard existing
107 data **((S3))**. When the data model is changed, for example due to new machines, protocols
108 or evolving research questions, the existing data must remain valid and usable. A change
109 in ontological semantics *now* must be compatible with previous semantics *then*.

110 **(R3) Searchability** The RDMS should have easily accessible search options not only for prop-
111 erty values of stored entities, but also for links to other entities and properties (and link)
112 thereof. This deep search allows the traversal of the structured knowledge graph and
113 delivers actual utility value.

114 **(R4) Sustainability** In order to assure long-term access to stored data, software solutions must
115 have some safeguard against becoming unmaintained. This could be achieved by being
116 either open-source software or “too big to fail”. In the case of open-source software,
117 either the community or other companies could step in, if the original maintainers stopped
118 their support. On the other hand, if a software system is very widely adopted and thus
119 indispensable, it is unlikely to be abandoned or left unsupported.

120 **(R5) Open APIs** For interaction with third-party programs, the RDM must have an API with
121 low entrance barriers ((S2)). In research contexts, these third-party programs are often
122 custom-written by scientists without explicit computer science background, so extensive
123 documentation of the API is very desirable.

124 3.2 Automation

125 Automation of repetitive data integration reduces error rates and frees users to concentrate on
126 more challenging tasks. It is therefore desirable for an RDMS to have:

127 **(R6) Synchronization** The RDMS should make it easy for its administrators to integrate existing
128 data sources (for example databases or file systems with structured folder hierarchies)
129 into the RDMS: The RDMS should be synchronized automatically with data from these
130 sources, which makes these data available in a unified manner via the RDMS interface.
131 Note that the RDMS can not solve the conceptual problem of a single source of truth when
132 synchronizing data from different sources, but it can at least highlight potential conflicts
133 and where they first occurred to administrators.

134 **(R7) ELN integration** Research work in the lab is increasingly documented with electronic lab
135 notebooks (ELNs)[24], [25], which allow to conveniently enter device and experimental
136 settings in a semi-structured way. This data is usually critical in the analysis of acquired
137 raw data from instruments, e.g., for searching specific data sets or filtering by parameters.
138 There should be a possibility that the RDMS integrates the ELN data and presents it like
139 data from other sources.

140 **(R8) Workflow representation** While following one SOP, the laboratory workflow is often
141 highly standardized, which makes it suitable for representation within the RDMS. The
142 RDMS should support workflows with different states, which can only be switched in an
143 admin-defined pattern. This simplifies the work for users, because they may e.g., only see
144 the interfaces which are relevant for the current sample processing step.

145 3.3 Specific requirements for scientific work

146 As introduced in section [Specifics of scientific research data management](#), some requirements
147 arise from scientific research specifically.

148 **(R9) Versioning** Mistakes during data acquisition happen, and it must be possible to correct
149 existing data sets. At the same time, this editing must be made transparent and the history
150 of each data set must be kept for future inspection.

151 **(R10) File system integration** For interaction with third-party programs, raw data files must be

152 available on standard file systems ((S1)). Ideally the scientists' workflows should remain
153 unchanged by the RDMS.

154 **(R11) Gentle learning curve, early pay-off** To accommodate for the short employment lifecy-
155 cles in science, RDMS should offer straightforward and simple to learn usage possibilities
156 which give some early sense of achievement ((S4), (S5))[26]. One example could be
157 simplified search options which help users understand that an RDMS will make their work
158 easier when handling with data.

159 3.4 Relation to FAIR data management

160 FAIR data management is seen as a general requirement by the scientific community at large.
161 We hold that a research data management system fulfilling (R1) – (R11) can enable research
162 groups to implement a FAIR data management.

163 Specifically, *Findability* can be achieved because each data set and collections of data can be
164 assigned persistent identifiers, data and metadata can be intimately connected and data can be
165 found through the search functionality of the RDMS.

166 Scientific RDMS can enable *Accessibility* through open and standardized APIs and separation of
167 raw data and metadata. RDMS allow for *Interoperability* when users can incorporate existing
168 ontologies for data model, descriptions and references between data sets. *Reusability* is fostered
169 by rich data models including licenses, provenance information and which follow the respective
170 communities' standards.

171 4 Current state of the tools landscape

172 We give a short overview over existing solutions, tools and approaches and over their possibilities.
173 We also classify the extent to which they cover the required features.

174 4.1 Technologies and approaches

175 Currently, DMS tools exist for a range of fields and use numerous technological and methodolog-
176 ical approaches. Different sources use different definitions for some of the following categories,
177 so we try make our definitions explicit, where necessary.

178 **ELNs** Here we use the definition of Harvard Medical School[27]:

179 An Electronic Lab Notebook (ELN) is a software tool that in its most basic
180 form replicates an interface much like a page in a paper lab notebook. In an
181 ELN you can enter protocols, observations, notes, and other data using your
182 computer or mobile device.

183 Electronic laboratory notebooks replace paper-based physical solutions to document the
184 scientific workflow in laboratories, but also partly planning and analysis of obtained data.
185 For the sake of this article, ELNs are distinct from other lab-oriented data management
186 software in that ELNs focus on the user experience while entering data in a laboratory envi-
187 ronment and allow to enter data in a semi-structured way, often much like a text editor with
188 the possibily to add a number of structured fields. The structure can sometimes be defin

189 by means of user editable templates. Typical examples are eLabFTW[28], Chemotion[29],
190 RSpace[30], eLabJournal[31] and other[24].

191 **Field-specific solutions** Many scientific field have specialized data management solutions for
192 their fields which cater to the specific needs, such as chemical structure searches, material
193 property tables, sample management or domain specific data visualization. Often, these
194 solutions excel in their purposes but customization options or interaction possibilities
195 may be limited. Examples are Nomad[32], C6H6.org[33], Chemotion[29], JuliaBase[34]
196 among others.

197 **Data, article and software repositories** Most scientific journals and some funding agencies
198 require scientists to publish the data underlying their publications in a publicly accessible
199 data repository. There are data repositories with custom software, and an increasing number
200 of public repository instances using off-the-rack software like Dataverse[35], Invenio[36],
201 DSpace[37] or CKAN[38]. Data repositories cover **(D1)** in the data lifecycle and offer
202 some search functionality, in all but very few cases they are intended for immutable data
203 at the time of publication. Data models range from very simple (only authors and text
204 description) over completely user defined key-value pairs to domain specific fixed data
205 models for domain repositories.

206 Similarly, software and articles are stored in specialized repositories, which often have
207 extensive metadata capabilities for the entities stored within them.

208 **Data storage systems** Data storage is a necessary prerequisite for scientific research and thus
209 there are many well established systems: mirrored network file systems (e.g., NFS, CIFS)
210 with regular backups to tape archives on the one hand and object stores (e.g., S3) on the
211 other hand, which store binary blobs outside classical file system structures.

212 **SQL databases** Plain SQL databases use tables where rows represent records and columns
213 represent the data sets' attributes or properties. Each table with a fixed set of columns of
214 mostly fixed types represent one type or class of data, defining the properties available for
215 that type.

216 Because SQL databases are readily available and can be integrated into most programming
217 languages, they are often used as the technical base for both self-written ad-hoc data
218 management solutions and existing commercial data management systems alike[39], [40].

219 **Key-value stores** A contrasting approach to SQL databases (therefore categorized as NoSQL
220 databases, popular examples are CouchDB or MongoDB), key-value stores manage data as
221 a collection of key-value pairs. They trade the structure of the SQL paradigm for flexibility,
222 allowing users to store whatever they deem appropriate.

223 **RDF, SPARQL** A common concept from academic knowledge representation research, RDF
224 (resource description framework)[41] is a framework and representation standard for
225 subject-predicate-object triples. It has found adoption in the standardization community
226 and some applications. SPARQL is a query language for accessing RDF data and used by
227 knowledge services such as Wikidata.[42], [43]

228 We would also like to mention that some solutions incorporate one or more of these approaches as

229 components. For example Kadi4Mat[44] and Nomad have ELNs as part of the overall software
230 package.

231 4.2 Do existing tools meet the requirements?

232 We discuss to what degree these technologies and tools are able to fulfill the requirements **(R1)**
233 – **(R11)** listed above. Here we differentiate between technologies, which may be used when
234 implementing applications on the one hand, and tools on the other hand which are candidates for
235 data management solutions.

236 4.2.1 Technologies

237 **RDF, SPARQL** RDF was designed and is well suited to represent semantic relationships be-
238 tween entities and local RDF collections can be extensively searched with SPARQL by
239 trained experts. There is a number of standardized RDF serializations which can be gen-
240 erated and read by a many programming languages. Data models can be implemented
241 using *RDF Schema*, which is based upon RDF. Entities can reference entities located on
242 other instances, which brings greater flexibility, but raises issues about data mutability
243 and searchability.

244 **SQL databases** Relational databases thrive on relations between tables and thus allow some
245 degree of semantic linking, albeit with very limited flexibility. Searching is possible, but
246 requires a certain degree of expertise, which can be mitigated by external helper tools.
247 There are standardized implementations, open source and proprietary alike, which can be
248 expected to continue for the foreseeable future.

249 **Key-value stores** NoSQL databases allow users a comprehensive degree of freedom when
250 storing data, but at the same time often provide no overarching structure to enforce certain
251 data model properties. Semantic linkage thus often is limited to convention instead of
252 internalized structures. Searchability is comparable to traditional SQL databases, and there
253 is a large number of implementations.

254 **Data storage systems** As a basic technology to store raw files or objects, data storage systems
255 do not have the ability to link data or provide a data model. Searching data for associated
256 metadata or file content is possible for some storage systems. Higher-level functionality is
257 not available within the data storage systems themselves.

258 These base technologies have in common that they mostly do not provide functionality such
259 as high-level network APIs, graphical user interfaces, integration with other components or
260 versioning. Also they target technical audiences and thus feature steep learning curves for data
261 manipulation and searching alike.

262 4.2.2 Tools

263 **ELNs** ELNs target at a non-technical audience and thus generally aim to have low entrance
264 barriers, with tutorials and graphical help functions. Most generic ELNs allow basic
265 linking between stored records and searches thereof, and users are guided in their work
266 of entering data by means of templates. These templates often do not have a semantic

Technology	R1	R2	R3	R4	R5	R6	R7	R8	R9	R10	R11
RDF + SPARQL	●	●	●	●	●	○	∅	∅	○	○	○
SQL	●	○	●	●	●	○	∅	∅	○	○	○
Key-value stores	◐	◐	◐	●	●	○	∅	∅	○	○	○
Data storage	○	○	◐	●	●	∅	∅	∅	◐	●	◐
Tools	R1	R2	R3	R4	R5	R6	R7	R8	R9	R10	R11
ELNs	◐	◐	◐	●	●	◐	∅	●	●	◐	●
FSS ^a	◐	◐	◐	◐	◐	◐	◐	●	◐	◐	●
Repositories	○	○	◐	●	◐	◐	○	○	●	○	●

a. field-specific solutions

R1	Semantic linkage	R7	ELN integration
R2	Flexible data model	R8	Workflow representation
R3	Searchability	R9	Versioning
R4	Sustainability	R10	File system integration
R5	Open APIs	R11	Gentle learning curve, early pay-off
R6	Synchronization		

Table 1: Data technologies, tools and if they meet the requirements.

Symbols used: ●: yes, ○: no, ◐: partly, ◑: may be possible to implement, ∅: not applicable.

Note that a “◐” may signify that not all particular examples of the category fulfill the requirement, but it may also mean that (nearly) all examples fulfill parts of the requirement.

267 meaning however, but serve only as a means of suggesting data fields. Data is organized
 268 around lab sessions, the main datatype are notes from the laboratory. ELNs only started to
 269 become the de-facto standard in laboratories over the last decade, so the market is far from
 270 settled. There are open-source and proprietary software solutions, by large players and by
 271 solo enterprises. Nearly all ELNs developed over the last five years now offer APIs for
 272 third-party access, and many allow users to organize their workflows, such as different
 273 processing steps for a sample.

274 Synchronization with other data sources or integration with file systems is not a core
 275 element of ELNs and as such rarely seen. Similarly, synchronization with other data
 276 sources exists only on a case-to-case base. Versioning of stored entities is possible to some
 277 extent for most ELNs.

278 **Field-specific solutions** Semantic linking may be possible to a certain amount as permitted by
 279 the data model, which typically is limited to the use cases foreseen by the developers.
 280 Similarly, searching the data often is limited to key-value filters on the specialized data
 281 types. Some solutions (e.g., NOMAD) implement their own ELNs, but integration with
 282 third-party ELNs and synchronization with other data sources does not exist generally: it
 283 could be implemented via APIs, in those cases where they exist. Support for workflows is
 284 generally quite good, and the learning curves are adapted to the audience. Versioning of
 285 data and integration of existing file systems may be present in some systems. Long-term
 286 availability of software support may be an issue when these solutions are only developed
 287 by a small set of people or even individuals, often in time-limited funding situations. In
 288 these cases, open-source software can be an insurance for the future, especially if there is

289 sufficient development documentation.

290 **Repositories** Data repositories only cover a small subset of data management use cases and as
291 such generally do not implement many of the requirements. They may allow semantic
292 linkage between entities, but do not have encompassing data models at all. Searching is
293 limited to key-value filters and full-text, sometimes referenced data sets can also be used
294 as filters, but there may be APIs which allow external tools to improve on this shortcoming.
295 Repositories generally have institutional funding so that long-term availability can be
296 seen as guaranteed. Synchronization with other data sources, local file system or ELN
297 integration or workflow representation does not make sense, since repositories are meant
298 for manual data archive upload at the end of the scientific life cycle. Upload of data
299 to archives is very straightforward in most cases, and editing of uploaded data does not
300 invalidate the original version, but only marks it as out of date.

301 4.2.3 Summary of existing tools

302 The requirements coverage of the examined technology and tool classes are shown in table 4.2.1.
303 We see that while existing tools cover a wide range of the required features, there are significant
304 shortcomings in two areas: flexible data models, semantic linkage and searchability on the one
305 hand, and integration with ELNs, other devices and file systems on the other hand. Due to
306 the large number of available products, for each requirement, there are ELN and field-specific
307 solutions which may fulfill it at least partly, although such a product in general does not cover
308 all requirements.

309 We stated earlier that these topical fields are especially relevant in scientific research. As an effect
310 DMS have been widely successful in many areas such as finance, administration, and high-tech
311 industries[45], [46], but remain scarce in both academic and private sector research[46], [47].
312 In summary, we find the need for a tool which fills the requirements for semantic, flexible data
313 management and has sufficient synchronization and ELN integration capabilities.

314 5 LinkAhead

315 We hold that LinkAhead[9], an agile data management framework, fulfills the proposed require-
316 ments from section Requirements for a scientific RDMS. LinkAhead was initially developed
317 under the name “CaosDB” by one of our colleagues, Timm Fitschen, during his time at the Max
318 Planck Institute for Dynamics and Self-Organization, and others.[48] In 2018, LinkAhead was
319 released, still as “CaosDB” under the AGPLv3 license on gitlab.com.[8] Since 2020, LinkAhead
320 has found increased adoption in multiple research facilities.

321 In this section, we first describe LinkAhead in detail, then we assess to which extent LinkAhead
322 fulfills the proposed requirements and finally we give an overview over limitations and possible
323 enhancements in the future.

324 5.1 Detailed description

325 LinkAhead was developed out of the need for a data management solution that can cope with
326 large data amount from automated sources and from existing file systems alike and that allows

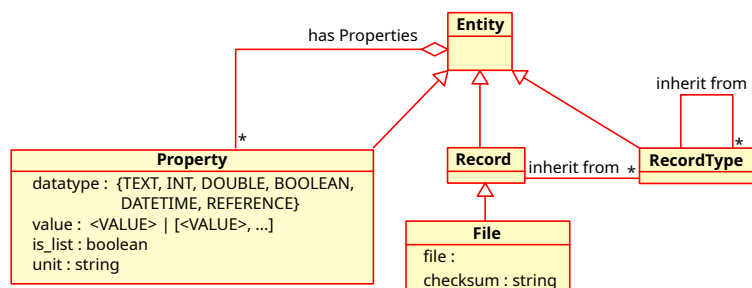


Figure 2: The metadata model of LinkAhead.

327 researchers to quickly adapt the way how data sets are connected or described. These needs
 328 reflect on the design choices which were taken over the course of development. LinkAhead
 329 is a general research data management system: specialized solutions such as ELNs, sample
 330 management systems, document management systems or other can be developed on top of it,
 331 according to specific needs.

332 5.1.1 Data Model

333 LinkAhead's *meta* data model is shown schematically in Figure 2. The base type for everything
 334 is ENTITY, with the inheriting types PROPERTY (attributes of ENTITIES, may be list values and
 335 references to other ENTITIES), RECORDTYPE (templates for actual data sets) and RECORD. Actual
 336 data is typically stored in RECORDS, which *inherit* from one or more RECORDTYPES and thus
 337 have all the PROPERTIES defined therein. The RECORDTYPES may form a complex inheritance
 338 hierarchy themselves. FILE entities are similar to Records, but additionally are connected to files
 339 which may reside on conventional file systems or potentially in abstracted cloud storage systems.
 340 This approach to use files at their current locations instead of duplicating file content not only
 341 increases LinkAhead's scalability, but also lower the entrance barrier, since scientists can access
 342 the managed file in their traditional ways.

343 Details of this metadata model in LinkAhead are elaborated on in [9], but it should be clear now
 344 already that LinkAhead provides the *Semantic linkage* feature.

345 In LinkAhead, the *data model* of the stored data refers to the RECORDTYPES and their PROPERTIES,
 346 which together describe the pattern to which newly created data sets should conform. The data
 347 model in LinkAhead can be modified at any time, but the changes only take effect for data to
 348 be inserted *after* this modification. Existing data is not affected and remains unchanged. This
 349 property fulfills the proposed *Flexible data model* feature.

350 PROPERTIES of RECORDTYPES are allocated a graded *importance*, which denotes if this PROPERTY
 351 is either *obligatory*, *recommended* or merely *suggested* for RECORDS which inherit from this
 352 RECORDTYPE, when a user creates a new RECORDS. This system of importances and the fact
 353 that *legacy* data is not necessarily consistent with a *modified* data model was a deliberate design
 354 decision. The rationale was that when the data model changes, the meaning at the time of data
 355 creation should have priority over consistency with later data models.

356 This possibility to completely change the data model, while not giving up on a general structure,
 357 places LinkAhead between traditional SQL based relational databases and NoSQL approaches

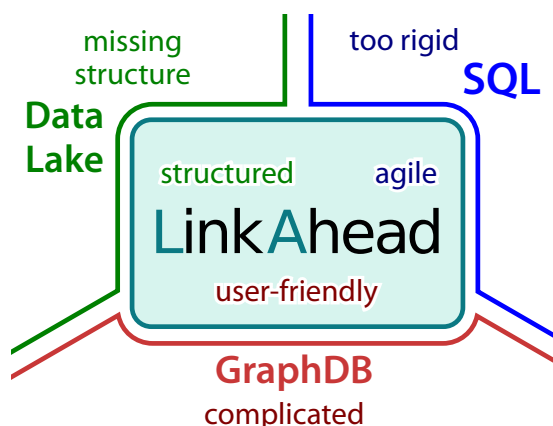


Figure 3: LinkAhead compared to other database approaches.

358 (c.f. Figure 3). While we described above why rigid SQL databases are not suited for use in
 359 dynamic research environments, giving no structure (the NoSQL paradigm) tends to lead to
 360 incoherent data which is hard to search. A common implementation of NoSQL approaches in
 361 the context of data management are *data lakes*, where raw data can be stored and annotated
 362 with metadata. The missing structure in Data Lakes however has led to the tongue-in-cheek
 363 colloquialism “Data Swamp”. A third approach, using graph databases to represent semantic
 364 information, has not found its way into general adoption to our knowledge, presumably because
 365 the query languages tend to become very unwieldy, compare the appendix [Appendix: Query](#)
 366 [language comparison](#) for an example.

367 5.1.2 Architecture and Libraries

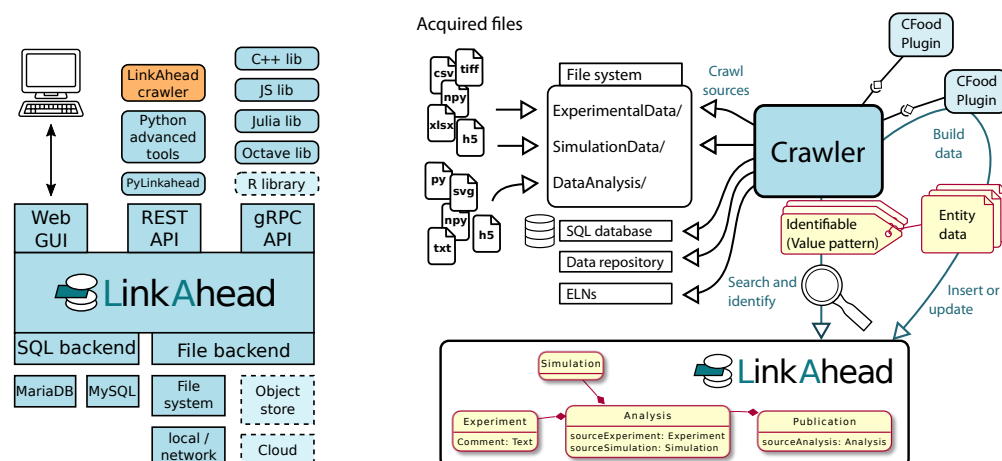


Figure 4: (a) LinkAhead’s server-client architecture with client libraries and backend components. Dotted elements are under development. (b) The crawler framework facilitates fast development of custom data integration from a diversity of sources.

368 LinkAhead uses a client/server based architecture, as depicted in Figure 4a. LinkAhead has is a
 369 REST API for simple access by traditional clients and a web interface for browsers, as well as a
 370 gRPC API which allows for more complex operations, such as atomic content manipulations. The

371 existing client libraries¹ and the open APIs provide the proposed *Interoperability* requirement.

372 One particularly useful client library component is the *LinkAhead Crawler* framework. This
 373 extensible framework simplifies the work to synchronize external data sources with LinkAhead
 374 through a plugin system. The crawler workflow can be characterized as follows:

- 375 1. The crawler checks its data sources for new or changed data stores, such as file systems or
 376 the content of other databases. This may happen periodically or be triggered manually by
 377 users.
- 378 2. Each new data source is fed to a so-called *CFood plugin* for consumption. There is a
 379 choice of existing plugins, or administrators can write their own. The CFood plugin's
 380 job is to build LinkAhead entities from the consumed data and to specify *Identifiables*,
 381 which work as search patterns. Administrators can mostly define simple CFood plugins by
 382 YAML configuration files[49] which is a more user-friendly approach than for example
 383 the mappings defined by the W3C's R2RML standard.[50]
- 384 3. The crawler checks for each *Identifiable* if a corresponding entity exists already in LinkA-
 385 head. If there is no corresponding entity, the entity as returned by the CFood plugin is
 386 inserted into LinkAhead. If there is already an existing entity, the Crawler will attempt
 387 to merge the existing with the new entity and notify the data curators in case of merge
 388 conflicts.

389 This tool set provides the *Synchronization* requirement, and if ELNs are used as external data
 390 source, the *ELN integration*. Practical use of LinkAhead crawler framework has previously
 391 been demonstrated in [51] and ELN integration was implemented as a working proof-of-concept
 392 in [52].

393 5.1.3 Miscellaneous features

394 **Deep search** LinkAhead offers a simple semantic query language, which borrows some seman-
 395 tics from SQL, but has a focus on usability for non-technical users. The LinkAhead query
 396 language makes deep search easy with expressions like the following:

```
397 FIND Analysis WITH quality_factor > 0.5
398         AND WITH Sample WITH weight < 80g
```

399 This convenient nesting of query expressions circumvents the JOIN operations from
 400 traditional SQL languages. A full documentation of LinkAhead's query language is
 401 available online[53] and in LinkAhead's sources.

402 **Search templates** LinkAhead's web interface provides customizable search templates which
 403 allow more advanced users to create their own query templates, which can then be shared
 404 with novice users for *simplified searches*. In query templates, users can insert custom
 405 strings into pre-defined locations of a search query, see Figure 5.

406 **Versioning** When entities are modified in LinkAhead, time and user of the change are recorded
 407 and LinkAhead puts the previous version onto a history stack and amends the current

1. A list of the available libraries with the respective source code repositories are given in the Appendix section [List of LinkAhead libraries](#).

408 version with link to the previous version. Over time, each entity may thus grow to a tree
 409 of linked versions, which can be retrieved via the web UI or programmatically through the
 410 APIs. This feature of LinkAhead enables scientific research data management users to
 411 adhere to the principles of good scientific practice.

412 **State management** In LinkAhead, users may declare a state machine of states and allowed
 413 transitions. Users may then affix states to entities, and these states can then only be
 414 changed according to the rules of the state machine. In this way, users can implement a
 415 *workflow representation* which ensure that for example laboratory samples run through a
 416 specified list of preparation steps in order.

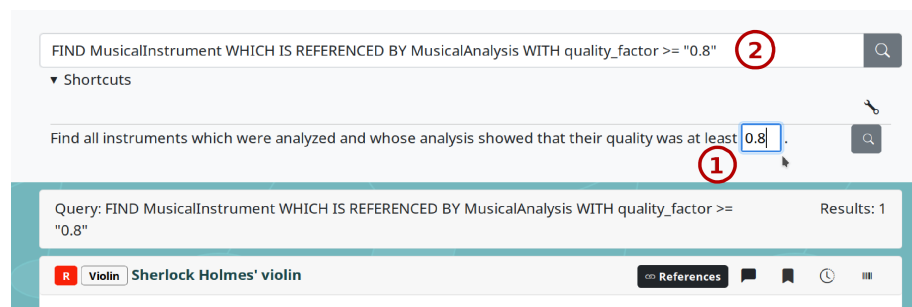


Figure 5: A query template in LinkAhead's web UI. The user can enter a custom value into an input field ① and the template is then executed as a plain LinkAhead query ②. Screenshot from <https://demo.indiscale.com>.

417 5.1.4 Availability and documentation

418 LinkAhead is available on the public Git repository gitlab.com at <https://gitlab.com/linkahead>, a detailed list of LinkAhead's sub projects is given in the annex. LinkAhead's
 419 source code is licensed under the AGPLv3 (Affero GNU Public License, version 3). Community
 420 contribution workflows, a code of conduct and general development guidelines are outlined in
 421 <https://gitlab.com/linkahead/linkahead-meta> and in the sub project specific code
 422 repositories. The community chat[54] is currently populated with 33 members, the contributors
 423 file lists 19 active contributors[48].

425 For the interested public, there is a live demo server at <https://demo.indiscale.com>,
 426 hosted by IndiScale GmbH. IndiScale GmbH also provides commercial support, development
 427 and customization services for LinkAhead. There are also Debian/Ubuntu packages to run
 428 precompiled LinkAhead for download at <https://indiscale.com/download>.

429 LinkAhead's sub projects each have their own documentation in their source directories. The
 430 documentation is also available online at <https://docs.indiscale.com>.

431 5.2 Requirements matching

432 In the following list, we evaluate if and how LinkAhead matches the requirements proposed in
 433 section [Requirements for a scientific RDMS](#):

434 **(R1) Semantic linkage** Links between ENTITIES in LinkAhead are implemented as reference
 435 typed PROPERTIES, these PROPERTIES can be restricted to Entities with certain parents,

436 adding an additional ontological level. All `PROPERTIES` can have a description and higher-
437 order properties and thus can fulfill the requirements for typical predicates in subject-
438 predicate-object relationships in predicate logic oriented triple stored such as RDF.

439 **(R2) Flexible data model** In LinkAhead, the data model, i.e., the set of `RECORDTYPES` can be
440 modified at any time. Existing `RECORDS` are not affected by these modifications and keep
441 their properties and inheritance information.

442 **(R3) Searchability** LinkAhead's query language allows to deeply search the available data for
443 simple key-value relations and also for nested relations on the knowledge graph and the
444 related entities' properties.

445 **(R4) Sustainability** LinkAhead is fully open-source and freely available on gitlab.com, with
446 options for commercial support.

447 **(R5) Open APIs** The REST and GRPC APIs included in LinkAhead enable interaction with
448 scientists' custom-written programs. Additionally the existence of client libraries simplifies
449 the usage by programmers without formal software development training.

450 **(R6) Synchronization** LinkAhead's *crawler* framework simplifies the synchronization between
451 existing data sources and the RDMS and allows to make a diversity of data accessible at a
452 single resource.

453 **(R7) ELN integration** The LinkAhead crawler may use ELNs as a data source, thus integrating
454 the content acquired by ELNs into the RDMS. This makes ELN data searchable and usable
455 equivalently to data from their sources.

456 **(R8) Workflow representation** The state machine in LinkAhead can be used to represent stan-
457 dardized workflows. For example laboratory samples or interview partners or publications
458 may have a state whose possible transitions and conditions can be specified.

459 **(R9) Versioning** Entities in LinkAhead are versioned and previous content may be displayed
460 and recovered. The content history of entities is stored: which user changed what value at
461 which time.

462 **(R10) File system integration** LinkAhead does not make copies of data files but only references
463 the file locations. The file path or resource identifier is returned upon queries, so that users
464 can use the location in their accustomed software.

465 **(R11) Gentle learning curve, early pay-off** Search queries in LinkAhead can be made more
466 accessible to users by templates where only specific values need to be filled in. The agile
467 data model allows scientists to start with a structured data management without the need to
468 develop a seemingly overwhelming master plan for their data. Instead they can start small
469 in an area where they expect the most immediate benefits such as improved findability of
470 linked data, and grow the data management at a later time.

471 We find that LinkAhead fulfills the requirements **(R1)–(R5)**, **(R9)–(R11)** “out of the box” and
472 that **(R6)–(R8)** (synchronization, ELN integration and workflows) can be readily implemented
473 using on-board means. LinkAhead therefore qualifies as a promising candidate for a scientific
474 RDMS.

475 5.3 Critical evaluation and outlook

476 A common misunderstanding about LinkAhead is what it provides out of the box and what it can
477 be used for. LinkAhead is not a tool to describe data objects following a specific ontology, but
478 ontologies can be implemented with LinkAhead in a straightforward manner, and it makes it easy
479 to manage data according to that ontology. It is not an ELN either: ELNs focus on unintrusive
480 interfaces for manual data acquisition, but mostly leaving handling of data from other sources, or
481 semantic data searches, to other tools. LinkAhead can be seen as a perfect complement to ELNs,
482 its primary goal is to make searching and linking of data beneficial for its users and to allow
483 for automation of all tasks. One data source for this automation may be ELNs, but of course
484 also other scientific data acquisition appliances such as laboratory hardware, high-performance
485 clusters or data repositories.

486 Similarly, LinkAhead does not enforce data to be FAIR. However researchers can use LinkAhead
487 to implement a FAIR data management and to assure that they handle their data in a FAIR manner.
488 Data transferred over the REST and GRPC interfaces use standardized formats such as XML for
489 data serialization, which can be understood by most programming interfaces. Additionally, the
490 internal infrastructure of LinkAhead is being reworked to use UUIDs or other unique identifiers
491 as primary keys for all ENTITIES.

492 As outlined in the previous section, LinkAhead fulfills most of the requirements and makes
493 others feasible for administrators and users. This also implies that there is room for improvement,
494 for example by providing integrated connectors to ELNs or other data sources or templates for
495 workflow representations.

496 Along similar lines, LinkAhead is still lacking tools to seamlessly interchange data and data
497 models with RDF based systems. In order to accelerate the general interoperability between data
498 management tools, LinkAhead has become part of the *ELN consortium*[55], an association of
499 interested parties with the aim to develop a common interchange format, based upon the RO-
500 Crate[56], [57] specification. While it is possible now already by external tools, full integration of
501 existing vocabularies represented in RDF serializations will further simplify FAIR data handling
502 with LinkAhead.

503 When synchronizing data with LinkAhead, special attention has to be given to the relationship
504 between data from external sources (e.g., crawled files, ELNs) and records in the RDMS. Different
505 sources can (usually by some error) have conflicting data, or entries in the RDMS can be changed
506 manually by users after their insertion. In our experience, this problem can not be solved in a
507 general and purely technical way. Instead, best practices have to be implemented as to where
508 possible errors should be corrected and whether some sources have precedence above each other.
509 An RDMS like LinkAhead, together with the crawler framework, can help administrators identify
510 inconsistencies in the case of two or more data sources. Through versioning, it is visible who
511 and when maybe changed data manually. How to optimize the help in recognizing potential
512 conflicts, and in the end curate data both in the RDMS and in the external sources, is subject of
513 the authors' ongoing research.

514 Since LinkAhead does not receive institutional funding, the direction of its future development
515 depends on the actions of the community. Therefore the immediate advancements will be shaped

516 by the needs of the current users of LinkAhead and of the company which currently provides
517 commercial support for it. A current list of feature requests can be generated online.[58] The
518 authors know of about a dozen institutions where LinkAhead is currently in use. Together with
519 the growing user base we expect the software to persist for a significant amount of time.

520 LinkAhead may fall short in terms of performance against traditional SQL databases for very
521 large amounts of data. To address this issue there is currently development underway to add a
522 virtualization layer which may use existing tabular data sources and present them in a configurable
523 way as native LinkAhead ENTITIES.[59]

524 We are aware that the perceived “usability” is subject to personal preferences unless evaluated in
525 a controlled study. We see the potential for a separate survey in the future which systematically
526 evaluates user experiences, workflows and the time and effort spent or gained by users of different
527 software approaches to a previously defined set of data management challenges.

528 6 Conclusion

529 We found that scientific research has specific needs to data management: Interoperability, agility,
530 adequate learning curves and early practical use. Altogether we identified a set of eleven
531 requirements which we applied to multiple classes of technologies and tools and to LinkAhead,
532 an agile RDMS. Especially in the requirements cluster “Semantic linkage, flexible data model,
533 semantic search”, previously existing tools show significant weaknesses, whereas LinkAhead
534 offers a promising outlook.

535 We hope that the open source license of LinkAhead will inspire more scientists to contribute to
536 LinkAhead and improve it in the areas of interoperability with existing standards.

537 7 Appendix: Software

538 7.1 LinkAhead

539 The LinkAhead suite with the main libraries is published at Zenodo:
540 <https://zenodo.org/record/7752417> (DOI:10.5281/zenodo.7752417)

541 7.2 List of LinkAhead libraries

542 The following libraries for programming client applications are publicly available:

543 **Python** <https://gitlab.com/linkahead/linkahead-pylib> The Python client library
544 can be used for third-party applications and is the foundation for several other libraries:

545 **Advanced Python tools** <https://gitlab.com/linkahead/linkahead-advanced-user-tools> Additional high-level tools building upon the Python library,
546 including a legacy implementation of the LinkAhead crawler. These tools also
547 include converters from JSON Schema to LinkAhead’s data model.
548

549 **Crawler** <https://gitlab.com/linkahead/linkahead-crawler> A new implemen-
550 tation of the LinkAhead crawler, also using the Python library. Allows to validate
551 data against a JSON Schema.

552 **JavaScript** <https://gitlab.com/linkahead/linkahead-webui> The JavaScript library
553 is part of the web user interface component.

554 **Protobuf API** <https://gitlab.com/linkahead/linkahead-protobuf> The gRPC API is
555 defined via these Protobuf files.

556 **C++** <https://gitlab.com/linkahead/linkahead-cpp-lib> The C++ library uses the
557 gRPC API of LinkAhead.

558 **Octave** <https://gitlab.com/linkahead/linkahead-octave-lib> The Octave/Matlab
559 library is based upon the C++ library.

560 **Julia** <https://gitlab.com/linkahead/linkahead-julia-lib> The Julia library also is
561 based upon the C++ library.

562 8 Appendix: Query language comparison

563 As an example for nested queries in different query languages, we consider the search for female
564 UK-based writers in a certain time period, whose given or family name starts with the letter
565 “M”. We used the RDF query language SPARQL with Wikidata (<https://www.wikidata.org>)
566 identifiers and LinkAhead’s query language with fictional but realistic identifier names.

567 The SPARQL query is as follows:


```
568 SELECT DISTINCT ?item ?itemLabel ?givenName ?familyName WHERE {
569     ?item wdt:P31 wd:Q5; # Any instance of a human.
570     wdt:P27 wd:Q145; # citizenship in the United Kingdom
571     wdt:P21 wd:Q6581072; # female
572     wdt:P106 wd:Q36180; # writer
573     wdt:P569 ?birthday;
574     wdt:P570 ?diedon;
575     wdt:P734 [rdfs:label ?familyName];
576     wdt:P735 [rdfs:label ?givenName].
577 FILTER(?birthday > "1870-01-01"^^xsd:dateTime
578     && ?diedon < "1950-01-01"^^xsd:dateTime)
579 FILTER(regex(?givenName, "M.*") || regex(?familyName, "M.*"))
580 SERVICE wikibase:label { bd:serviceParam wikibase:language "en" }
581 }
```

582 In contrast, the LinkAhead query looks like this:

```
583 SELECT given_name, family_name FROM Writer
584 WITH gender=f AND citizenship=UK AND birthday > 1870 AND death < 1950
585 AND (given_name LIKE "M*" OR family_name LIKE "M*")
```

586 We understand that SPARQL and LinkAhead’s query language have non-overlapping sets of
587 features. For example, LinkAhead does not know about aliases for names, such as in multilingual
588 environments. On the other hand, SPARQL has no native understanding of SI units and their
589 conversion and it focuses on experts instead of casual users.

590 9 Acknowledgements

591 We acknowledge the previous work on the LinkAhead software by its main authors and indepen-
592 dent contributors[48], especially Timm Fitschen .

593 10 Conflicts of interest

594 The authors work for IndiScale GmbH, which provides commercial support and other services
595 for LinkAhead. DH and FS contributed to the development of LinkAhead.

596 11 Roles and contributions

597 **Daniel Hornung:** Conceptualization, Visualization, Writing – original draft

598 **Florian Spreckelsen:** Conceptualization, Writing – review & editing

599 **Thomas Weiß:** Conceptualization, Visualization

600 References

- 601 [1] C. R. Bauer, N. Umbach, B. Baum, *et al.*, “Architecture of a Biomedical Informatics Re-
602 search Data Management Pipeline,” *Exploring Complexity in Health: An Interdisciplinary*
603 *Systems Approach*, pp. 262–266, 2016. doi: [10.3233/978-1-61499-678-1-262](https://doi.org/10.3233/978-1-61499-678-1-262).
604 [Online]. Available: [https://ebooks.iospress.nl/doi/10.3233/978-1-61499-6](https://ebooks.iospress.nl/doi/10.3233/978-1-61499-678-1-262)
605 [78-1-262](https://ebooks.iospress.nl/doi/10.3233/978-1-61499-678-1-262).
- 606 [2] C. L. Borgman, “The conundrum of sharing research data,” *Journal of the American*
607 *Society for Information Science and Technology*, vol. 63, no. 6, pp. 1059–1078, 2012,
608 issn: 1532-2890. doi: [10.1002/asi.22634](https://doi.org/10.1002/asi.22634). [Online]. Available: [https://onlinelib](https://onlinelibrary.wiley.com/doi/abs/10.1002/asi.22634)
609 [rary.wiley.com/doi/abs/10.1002/asi.22634](https://onlinelibrary.wiley.com/doi/abs/10.1002/asi.22634).
- 610 [3] A. M. Khattak, K. Latif, and S. Lee, “Change management in evolving web ontologies,”
611 *Knowledge-Based Systems*, vol. 37, pp. 1–18, Jan. 1, 2013, issn: 0950-7051. doi: [10.101](https://doi.org/10.1016/j.knosys.2012.05.005)
612 [6/j.knosys.2012.05.005](https://doi.org/10.1016/j.knosys.2012.05.005). [Online]. Available: [https://www.sciencedirect.com](https://www.sciencedirect.com/science/article/pii/S0950705112001323)
613 [/science/article/pii/S0950705112001323](https://www.sciencedirect.com/science/article/pii/S0950705112001323) (visited on 01/18/2023).
- 614 [4] T. Schneider and M. Šimkus, “Ontologies and Data Management: A Brief Survey,” *KI -*
615 *Künstliche Intelligenz*, vol. 34, Aug. 13, 2020. doi: [10.1007/s13218-020-00686-3](https://doi.org/10.1007/s13218-020-00686-3).
- 616 [5] M. D. Wilkinson, M. Dumontier, IJ. J. Aalbersberg, *et al.*, “The FAIR Guiding Principles
617 for scientific data management and stewardship,” *Scientific Data*, vol. 3, no. 1, p. 160 018,
618 1 Mar. 15, 2016, issn: 2052-4463. doi: [10.1038/sdata.2016.18](https://doi.org/10.1038/sdata.2016.18). [Online]. Available:
619 <https://www.nature.com/articles/sdata201618> (visited on 11/22/2021).
- 620 [6] R. Higman, D. Bangert, and S. Jones, “Three camps, one destination: The intersections of
621 research data management, FAIR and Open,” *Insights*, vol. 32, no. 1, p. 18, 1 May 22,
622 2019, issn: 2048-7754. doi: [10.1629/uksg.468](https://doi.org/10.1629/uksg.468). [Online]. Available: [http://insight](http://insights.uksg.org/articles/10.1629/uksg.468/)
623 [s.uksg.org/articles/10.1629/uksg.468/](http://insights.uksg.org/articles/10.1629/uksg.468/) (visited on 01/18/2023).

- 624 [7] G. W. Hruby, J. McKiernan, S. Bakken, and C. Weng, “A centralized research data repository enhances retrospective outcomes research capacity: A case report,” *Journal of the American Medical Informatics Association*, vol. 20, no. 3, pp. 563–567, May 1, 2013, issn: 1067-5027. doi: [10.1136/amiajnl-2012-001302](https://doi.org/10.1136/amiajnl-2012-001302). [Online]. Available: <https://doi.org/10.1136/amiajnl-2012-001302>.
- 625
626
627
628
- 629 [8] “LinkAhead.” Website: <https://getlinkahead.com>, Source code: <https://gitlab.com/linkahead>, GitLab. (Oct. 5, 2023).
- 630
- 631 [9] T. Fitschen, A. Schlemmer, D. Hornung, H. tom Wörden, U. Parlitz, and S. Luther, “CaosDB—Research Data Management for Complex, Changing, and Automated Research Workflows,” *Data*, vol. 4, no. 2, p. 83, 2 Jun. 2019, issn: 2306-5729. doi: [10.3390/data4020083](https://doi.org/10.3390/data4020083). [Online]. Available: <https://www.mdpi.com/2306-5729/4/2/83> (visited on 01/18/2023).
- 632
633
634
635
- 636 [10] F. Radicchi, S. Fortunato, and A. Vespignani, “Citation Networks,” in *Models of Science Dynamics: Encounters Between Complexity Theory and Information Sciences*, ser. Understanding Complex Systems, A. Scharnhorst, K. Börner, and P. van den Besselaar, Eds., Berlin, Heidelberg: Springer, 2012, pp. 233–257, isbn: 978-3-642-23068-4. doi: [10.1007/978-3-642-23068-4_7](https://doi.org/10.1007/978-3-642-23068-4_7). [Online]. Available: https://doi.org/10.1007/978-3-642-23068-4_7.
- 637
638
639
640
641
- 642 [11] K. Manghani, “Quality assurance: Importance of systems and standard operating procedures,” *Perspectives in Clinical Research*, vol. 2, no. 1, pp. 34–37, 2011, issn: 2229-3485. doi: [10.4103/2229-3485.76288](https://doi.org/10.4103/2229-3485.76288). pmid: 21584180. [Online]. Available: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3088954/> (visited on 07/11/2023).
- 643
644
645
- 646 [12] M. A. Abujayyab, M. Hamouda, and A. Aly Hassan, “Biological treatment of produced water: A comprehensive review and metadata analysis,” *Journal of Petroleum Science and Engineering*, vol. 209, p. 109914, Feb. 1, 2022, issn: 0920-4105. doi: [10.1016/j.petrol.2021.109914](https://doi.org/10.1016/j.petrol.2021.109914). [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0920410521015308> (visited on 07/11/2023).
- 647
648
649
650
- 651 [13] A. Nicholson, D. McIsaac, C. MacDonald, *et al.*, “An analysis of metadata reporting in freshwater environmental DNA research calls for the development of best practice guidelines,” *Environmental DNA*, vol. 2, no. 3, pp. 343–349, 2020, issn: 2637-4943. doi: [10.1002/edn3.81](https://doi.org/10.1002/edn3.81). [Online]. Available: <https://onlinelibrary.wiley.com/doi/abs/10.1002/edn3.81>.
- 652
653
654
655
- 656 [14] J. Griss, Y. Perez-Riverol, H. Hermjakob, and J. A. Vizcaíno, “Identifying novel biomarkers through data mining—A realistic scenario?” *PROTEOMICS—Clinical Applications*, vol. 9, no. 3-4, pp. 437–443, 2015, issn: 1862-8354. doi: [10.1002/prca.201400107](https://doi.org/10.1002/prca.201400107). [Online]. Available: <https://onlinelibrary.wiley.com/doi/abs/10.1002/prca.201400107>.
- 657
658
659
660
- 661 [15] G. Silvello, G. Bordea, N. Ferro, P. Buitelaar, and T. Bogers, “Semantic representation and enrichment of information retrieval experimental data,” *International Journal on Digital Libraries*, vol. 18, no. 2, pp. 145–172, Jun. 1, 2017, issn: 1432-1300. doi: [10.1007/s00799-016-0172-8](https://doi.org/10.1007/s00799-016-0172-8). [Online]. Available: <https://doi.org/10.1007/s00799-016-0172-8>.
- 662
663
664
665

- 666 [16] B. G. Fitzpatrick, “Issues in Reproducible Simulation Research,” *Bulletin of Mathematical*
667 *Biology*, vol. 81, no. 1, pp. 1–6, Jan. 1, 2019, issn: 1522-9602. doi: [10.1007/s11538-01](https://doi.org/10.1007/s11538-018-0496-1)
668 [8-0496-1](https://doi.org/10.1007/s11538-018-0496-1). [Online]. Available: <https://doi.org/10.1007/s11538-018-0496-1>.
- 669 [17] R. A. Poldrack, K. J. Gorgolewski, and G. Varoquaux, “Computational and Informatic
670 Advances for Reproducible Data Analysis in Neuroimaging,” *Annual Review of Biomedical*
671 *Data Science*, vol. 2, no. 1, pp. 119–138, 2019. doi: [10.1146/annurev-biodatasci-0](https://doi.org/10.1146/annurev-biodatasci-072018-021237)
672 [72018-021237](https://doi.org/10.1146/annurev-biodatasci-072018-021237). [Online]. Available: [https://doi.org/10.1146/annurev-biodata](https://doi.org/10.1146/annurev-biodatasci-072018-021237)
673 [sci-072018-021237](https://doi.org/10.1146/annurev-biodatasci-072018-021237).
- 674 [18] F. Strozzi, R. Janssen, R. Wurmus, *et al.*, “Scalable Workflows and Reproducible Data
675 Analysis for Genomics,” in *Evolutionary Genomics: Statistical and Computational Meth-*
676 *ods*, ser. Methods in Molecular Biology, M. Anisimova, Ed., New York, NY: Springer,
677 2019, pp. 723–745, isbn: 978-1-4939-9074-0. doi: [10.1007/978-1-4939-9074-0_24](https://doi.org/10.1007/978-1-4939-9074-0_24).
678 [Online]. Available: https://doi.org/10.1007/978-1-4939-9074-0_24.
- 679 [19] J. E. Hannay, C. MacLeod, J. Singer, H. P. Langtangen, D. Pfahl, and G. Wilson, “How
680 do scientists develop and use scientific software?” In *2009 ICSE Workshop on Software*
681 *Engineering for Computational Science and Engineering*, May 2009, pp. 1–8. doi: [10.11](https://doi.org/10.1109/SECSE.2009.5069155)
682 [09/SECSE.2009.5069155](https://doi.org/10.1109/SECSE.2009.5069155).
- 683 [20] J. Segal, “When Software Engineers Met Research Scientists: A Case Study,” *Empirical*
684 *Software Engineering*, vol. 10, no. 4, pp. 517–536, Oct. 1, 2005, issn: 1573-7616. doi:
685 [10.1007/s10664-005-3865-y](https://doi.org/10.1007/s10664-005-3865-y). [Online]. Available: [0664-005-3865-y](https://doi.org/10.1007/s1
686 <a href=).
- 687 [21] G. Wilson, “Software Carpentry: Getting Scientists to Write Better Code by Making Them
688 More Productive,” *Computing in Science & Engineering*, vol. 8, no. 6, pp. 66–69, Nov.
689 2006, issn: 1558-366X. doi: [10.1109/MCSE.2006.122](https://doi.org/10.1109/MCSE.2006.122).
- 690 [22] N. R. Anderson, E. S. Lee, J. S. Brockenbrough, *et al.*, “Issues in Biomedical Research
691 Data Management and Analysis: Needs and Barriers,” *Journal of the American Medical*
692 *Informatics Association*, vol. 14, no. 4, pp. 478–488, Jul. 1, 2007, issn: 1067-5027. doi:
693 [10.1197/jamia.M2114](https://doi.org/10.1197/jamia.M2114). [Online]. Available: [114](https://doi.org/10.1197/jamia.M2
694 <a href=).
- 695 [23] M. Bron, J. Van Gorp, and M. de Rijke, “Media studies research in the data-driven age:
696 How research questions evolve,” *Journal of the Association for Information Science and*
697 *Technology*, vol. 67, no. 7, pp. 1535–1554, 2016, issn: 2330-1643. doi: [10.1002/asi.23](https://doi.org/10.1002/asi.23458)
698 [458](https://doi.org/10.1002/asi.23458). [Online]. Available: [https://onlinelibrary.wiley.com/doi/abs/10.1002](https://onlinelibrary.wiley.com/doi/abs/10.1002/asi.23458)
699 [/asi.23458](https://onlinelibrary.wiley.com/doi/abs/10.1002/asi.23458).
- 700 [24] S. Kanza, C. Willoughby, N. Gibbins, *et al.*, “Electronic lab notebooks: Can they replace
701 paper?” *Journal of Cheminformatics*, vol. 9, no. 1, p. 31, May 24, 2017, issn: 1758-2946.
702 doi: [10.1186/s13321-017-0221-3](https://doi.org/10.1186/s13321-017-0221-3). [Online]. Available: [https://doi.org/10.118](https://doi.org/10.1186/s13321-017-0221-3)
703 [6/s13321-017-0221-3](https://doi.org/10.1186/s13321-017-0221-3).
- 704 [25] S. G. Higgins, A. A. Nogiwa-Valdez, and M. M. Stevens, “Considerations for implementing
705 electronic laboratory notebooks in an academic research environment,” *Nature Protocols*,
706 vol. 17, no. 2, pp. 179–189, 2 Feb. 2022, issn: 1750-2799. doi: [10.1038/s41596-021-0](https://doi.org/10.1038/s41596-021-0)

- 707 0645-8. [Online]. Available: <https://www.nature.com/articles/s41596-021-00>
708 645-8 (visited on 01/20/2023).
- 709 [26] F. Abdullah, R. Ward, and E. Ahmed, "Investigating the influence of the most commonly
710 used external variables of TAM on students' Perceived Ease of Use (PEOU) and Perceived
711 Usefulness (PU) of e-portfolios," *Computers in Human Behavior*, vol. 63, no. C, pp. 75–90,
712 Oct. 1, 2016, issn: 0747-5632. doi: [10.1016/j.chb.2016.05.014](https://doi.org/10.1016/j.chb.2016.05.014). [Online]. Available:
713 <https://doi.org/10.1016/j.chb.2016.05.014>.
- 714 [27] Harvard Medical School, Boston, MA. "Electronic Lab Notebooks." (2023), [Online].
715 Available: [https://datamanagement.hms.harvard.edu/collect-analyze/elec](https://datamanagement.hms.harvard.edu/collect-analyze/electronic-lab-notebooks)
716 [tronic-lab-notebooks](https://datamanagement.hms.harvard.edu/collect-analyze/electronic-lab-notebooks) (visited on 09/27/2023).
- 717 [28] N. Carpi, A. Minges, and M. Piel, "eLabFTW: An open source laboratory notebook for
718 research labs," *Journal of Open Source Software*, vol. 2, no. 12, p. 146, Apr. 14, 2017,
719 Sources: <https://github.com/elabftw/>, issn: 2475-9066. doi: [10.21105/joss.00](https://doi.org/10.21105/joss.0014146)
720 [146](https://doi.org/10.21105/joss.0014146). [Online]. Available: [https://joss.theoj.org/papers/10.21105/joss.0014](https://joss.theoj.org/papers/10.21105/joss.0014146)
721 [6](https://joss.theoj.org/papers/10.21105/joss.0014146).
- 722 [29] P. Tremouilhac, A. Nguyen, Y.-C. Huang, *et al.*, "Chemotion ELN: An Open Source
723 electronic lab notebook for chemists in academia," *Journal of Cheminformatics*, vol. 9,
724 no. 1, p. 54, Sep. 25, 2017, issn: 1758-2946. doi: [10.1186/s13321-017-0240-0](https://doi.org/10.1186/s13321-017-0240-0).
725 [Online]. Available: <https://doi.org/10.1186/s13321-017-0240-0>.
- 726 [30] *RSpace ELN*, Formerly eCAT from the University of Wisconsin., Edinburgh, Scotland:
727 Research Space. [Online]. Available: <https://www.researchspace.com/> (visited on
728 07/12/2023).
- 729 [31] *eLabJournal*, eLabNext. [Online]. Available: [https://www.elabnext.com/products](https://www.elabnext.com/products/eLabJournal/)
730 [/eLabJournal/](https://www.elabnext.com/products/eLabJournal/).
- 731 [32] C. Draxl and M. Scheffler, "The NOMAD laboratory: From data sharing to artificial
732 intelligence," *Journal of Physics: Materials*, vol. 2, no. 3, p. 036 001, May 2019, Sources:
733 <https://gitlab.mpcdf.mpg.de/nomad-lab/nomad-FAIR>, issn: 2515-7639. doi:
734 [10.1088/2515-7639/ab13bb](https://doi.org/10.1088/2515-7639/ab13bb). [Online]. Available: [https://dx.doi.org/10.1088](https://dx.doi.org/10.1088/2515-7639/ab13bb)
735 [/2515-7639/ab13bb](https://dx.doi.org/10.1088/2515-7639/ab13bb).
- 736 [33] L. Patiny, M. Zasso, D. Kostro, *et al.*, "The C6H6 NMR repository: An integral solution
737 to control the flow of your data from the magnet to the public," *Magnetic Resonance in*
738 *Chemistry*, vol. 56, no. 6, pp. 520–528, 2018, issn: 1097-458X. doi: [10.1002/mrc.4669](https://doi.org/10.1002/mrc.4669).
739 [Online]. Available: [https://onlinelibrary.wiley.com/doi/abs/10.1002/mrc](https://onlinelibrary.wiley.com/doi/abs/10.1002/mrc.4669)
740 [.4669](https://onlinelibrary.wiley.com/doi/abs/10.1002/mrc.4669).
- 741 [34] T. Bronger. "Introduction — JuliaBase, the samples database." (2023), [Online]. Available:
742 <https://www.juliabase.org/> (visited on 09/27/2023).
- 743 [35] G. King, "An Introduction to the Dataverse Network as an Infrastructure for Data Sharing,"
744 *Sociological Methods & Research*, vol. 36, no. 2, pp. 173–199, Nov. 1, 2007, issn: 0049-
745 1241. doi: [10.1177/0049124107306660](https://doi.org/10.1177/0049124107306660). [Online]. Available: [https://doi.org/10](https://doi.org/10.1177/0049124107306660)
746 [.1177/0049124107306660](https://doi.org/10.1177/0049124107306660).

- 747 [36] J. Caffaro and S. Kaplun. “Invenio: A Modern Digital Library for Grey Literature.”
748 Sources: <https://github.com/inveniosoftware>. (2010), [Online]. Available: <http://cds.cern.ch/record/1312678> (visited on 07/12/2023), preprint.
- 750 [37] M. Smith, M. Barton, M. Bass, *et al.*, “DSpace: An Open Source Dynamic Digital Repository,” Jan. 2003, issn: 1082-9873. doi: [10.1045/january2003-smith](https://doi.org/10.1045/january2003-smith). [Online]. Available: <https://dspace.mit.edu/handle/1721.1/29465> (visited on 07/12/2023).
- 753 [38] J. Winn, “Open data and the academy: An evaluation of CKAN for research data management,” presented at the IASSIST 2013, Cologne, May 2013. [Online]. Available: <http://eprints.lincoln.ac.uk/id/eprint/9778/> (visited on 07/12/2023).
- 756 [39] A. Ailamaki, V. Kantere, and D. Dash, “Managing scientific data,” *Communications of the ACM*, vol. 53, no. 6, pp. 68–78, Jun. 1, 2010, issn: 0001-0782. doi: [10.1145/1743546.1743568](https://doi.org/10.1145/1743546.1743568). [Online]. Available: <https://doi.org/10.1145/1743546.1743568>.
- 759 [40] A. Nourani, H. Ayatollahi, and M. S. Dodaran, “Clinical Trial Data Management Software: A Review of the Technical Features,” *Reviews on Recent Clinical Trials*, vol. 14, no. 3, pp. 160–172, Sep. 1, 2019. doi: [10.2174/1574887114666190207151500](https://doi.org/10.2174/1574887114666190207151500).
- 762 [41] O. Hartig, P.-A. Champin, and G. Kellogg. “RDF 1.2 Concepts and Abstract Syntax.” (Jun. 29, 2023), [Online]. Available: <https://www.w3.org/TR/2023/WD-rdf12-concepts-20230629/> (visited on 07/12/2023).
- 765 [42] S. Harris and A. Seaborne. “SPARQL 1.1 Query Language.” (Mar. 21, 2013), [Online]. Available: <https://www.w3.org/TR/2013/REC-sparql11-query-20130321/> (visited on 07/12/2023).
- 768 [43] D. Vrandečić and M. Krötzsch, “Wikidata: A free collaborative knowledgebase,” *Communications of the ACM*, vol. 57, no. 10, pp. 78–85, Sep. 23, 2014, issn: 0001-0782, 1557-7317. doi: [10.1145/2629489](https://doi.org/10.1145/2629489). [Online]. Available: <https://dl.acm.org/doi/10.1145/2629489>.
- 772 [44] N. Brandt, L. Griem, C. Herrmann, *et al.*, “Kadi4Mat: A Research Data Infrastructure for Materials Science,” vol. 20, no. 1, p. 8, 1 Feb. 10, 2021, issn: 1683-1470. doi: [10.5334/dsj-2021-008](https://doi.org/10.5334/dsj-2021-008). [Online]. Available: <https://datascience.codata.org/articles/10.5334/dsj-2021-008> (visited on 09/27/2023).
- 776 [45] T. P. Raptis, A. Passarella, and M. Conti, “Data Management in Industry 4.0: State of the Art and Open Challenges,” *IEEE Access*, vol. 7, pp. 97 052–97 093, Jul. 16, 2019, issn: 2169-3536. doi: [10.1109/ACCESS.2019.2929296](https://doi.org/10.1109/ACCESS.2019.2929296).
- 779 [46] S. V. Tuyl and A. Whitmire, “Investigation of Non-Academic Data Management Practices to Inform Academic Research Data Management,” *Research Ideas and Outcomes*, vol. 4, e30829, Oct. 31, 2018, issn: 2367-7163. doi: [10.3897/rio.4.e30829](https://doi.org/10.3897/rio.4.e30829). [Online]. Available: <https://riojournal.com/article/30829/> (visited on 01/19/2023).
- 783 [47] M. A. Kennan and L. Markauskaite, “Research Data Management Practices: A Snapshot in Time,” *International Journal of Digital Curation*, vol. 10, no. 2, pp. 69–95, 2 Jun. 30, 2015, issn: 1746-8256. doi: [10.2218/ijdc.v10i2.329](https://doi.org/10.2218/ijdc.v10i2.329). [Online]. Available: <http://ijdc.net/index.php/ijdc/article/view/10.2.69> (visited on 01/19/2023).

- 787 [48] LinkAhead. “HUMANS.md,” GitLab. (Sep. 13, 2023), [Online]. Available: <https://gitlab.com/linkahead/linkahead/-/blob/main/HUMANS.md> (visited on 10/05/2023).
- 788
- 789 [49] “Tutorial: Parameter file — LinkAhead Crawler 0.7.0 documentation.” (2023), [Online].
- 790 Available: <https://docs.indiscale.com/caosdb-crawler/tutorials/parameterfile.html#getting-started-with-the-cfood> (visited on 07/12/2023).
- 791
- 792 [50] “R2RML: RDB to RDF Mapping Language.” (Sep. 27, 2012), [Online]. Available: <http://www.w3.org/TR/2012/REC-r2rml-20120927/> (visited on 07/12/2023).
- 793
- 794 [51] F. Spreckelsen, B. Rüchardt, J. Lebert, S. Luther, U. Parlitz, and A. Schlemmer, “Guidelines
- 795 for a Standardized Filesystem Layout for Scientific Data,” *Data*, vol. 5, no. 2, p. 43, 2
- 796 Jun. 2020, issn: 2306-5729. doi: [10.3390/data5020043](https://doi.org/10.3390/data5020043). [Online]. Available: <https://www.mdpi.com/2306-5729/5/2/43> (visited on 01/18/2023).
- 797
- 798 [52] “LinkAhead / LinkAhead Crawler Cfoods / ELabFTW Cfood · GitLab,” GitLab. (Oct. 5,
- 799 2023), [Online]. Available: <https://gitlab.com/linkahead/crawler-extensions/elabftw-cfood> (visited on 10/05/2023).
- 800
- 801 [53] “LinkAhead Query Language Examples — linkahead-server 0.11.0 documentation.”
- 802 (2023), [Online]. Available: <https://docs.indiscale.com/caosdb-server/LinkAhead-Query-Language.html> (visited on 10/05/2023).
- 803
- 804 [54] “#Linkahead on Matrix chat.” (Oct. 5, 2023), [Online]. Available: <https://matrix.to/#/#linkahead:matrix.org> (visited on 10/05/2023).
- 805
- 806 [55] “The ELN Consortium,” GitHub. (2023), [Online]. Available: <https://github.com/TheELNConsortium> (visited on 07/12/2023).
- 807
- 808 [56] P. Sefton, E. Ó Carragáin, S. Soiland-Reyes, *et al.*, “RO-Crate Metadata Specification
- 809 1.1.3,” Apr. 26, 2023. doi: [10.5281/zenodo.7867028](https://doi.org/10.5281/zenodo.7867028). [Online]. Available: <https://zenodo.org/record/7867028> (visited on 07/12/2023).
- 810
- 811 [57] S. Soiland-Reyes, P. Sefton, M. Crosas, *et al.*, “Packaging research artefacts with RO-
- 812 Crate,” *Data Science*, vol. 5, no. 2, pp. 97–138, Jan. 1, 2022, issn: 2451-8484. doi: [10.3233/DS-210053](https://doi.org/10.3233/DS-210053). [Online]. Available: <https://content.iospress.com/articles/data-science/ds210053> (visited on 07/12/2023).
- 813
- 814
- 815 [58] “LinkAhead Enhancement Requests,” GitLab - LinkAhead. (Oct. 5, 2023), [Online].
- 816 Available: [https://gitlab.com/groups/linkahead/-/issues?or\[label_name\]\[\]=Enhancement%3A%3AAccepted&or\[label_name\]\[\]=Enhancement%3A%3AProposed&state=opened](https://gitlab.com/groups/linkahead/-/issues?or[label_name][]=Enhancement%3A%3AAccepted&or[label_name][]=Enhancement%3A%3AProposed&state=opened) (visited on 10/05/2023).
- 817
- 818
- 819 [59] “LinkAhead Trino branch,” GitLab LinkAhead. (Oct. 5, 2023), [Online]. Available: <https://gitlab.com/linkahead/linkahead-server/-/tree/f-experiment-trino>
- 820 (visited on 10/05/2023).
- 821