Proposing a CORD-19 SDK Solution to Improve Machine Readability

Supplemental to *Advances in Ubiquitous Computing: Cyber-Physical Systems, Smart Cities and Ecological Monitoring*, edited by Amy Neustein, part of the Series "Advances in Ubiquitous Sensing Applications for Healthcare", Elsevier, series editors Nilanjen Dey, Amira Ashour, and Simon James Fong.

The forthcoming volume *Advances in Ubiquitous Computing: Cyber-Physical Systems, Smart Cities and Ecological Monitoring* will prototype a data-publishing protocol that extends existing initiatives such as "Research Objects" and **FAIR** ("Findable, Accessible, Interoperable, Reusable"), both of which reflect publishers' current emphasis on publicly sharing research data. In the last few years, academic, governmental, and industry stakeholders have increasingly prioritized open-access data publication as a complement to the traditional ecosystem of peer-reviewed scientific literature. This new "data-centric" paradigm seeks to augment the value and accuracy research data by provisioning scientists with digital tools to find, evaluate, and reuse data which is relevant to their research. At the present moment, the urgency of sharing Covid-19 research is further boosting this data-centric and open-access publishing model. Indeed, Covid-19 presents a unique opportunity for researchers and publishers to assess the strengths and weaknesses of current publishing technology, because we are now in a global environment where there is an unprecedented societal demand for pooling and accelerating research in one specific, easily demarcated domain (that of Covid-19 in particular and coronaviruses in general).

Realizing a new data-centric publishing paradigm requires more than just enouraging authors to prepare downloadable data sets to accompanying their research papers. It also requires a technological infrastructure to support the curation of data sets and data set collections. New digital solutions are needed to accommodate the sheer volume and diversity of peer-reviewed research work, and the implementation of these solutions will have to be guided by interdisciplinary theoretical models integrating computer science, knowledge engineering, biblioinformatics, as well as individual natural and social sciences. In this vein, one goal of *Advances in Ubiquitous Computing* is to advocate for an even more rigorous data publishing protocol, taking a multi-disciplinary and multi-paradigm perspective on the curation and structuring of published research data. In particular, as a supplement to *Advances*, several data sets will be published (or republished) relevant to subjects covered by chapters in the text, such as signal-processing, bioacoustics, and Natural Language Processing. The primary purpose of these datasets is to demonstrate data-management and software-development techniques discussed in the volume, particularly in the third chapter (the data sets are being curated by that chapter's author). These data sets will introduce a so-called "**MOSAIC**" protocol which, compared to ordinary Research Objects, places greater emphasis on (1) self-contained code libraries bundled with data sets to provide sophisticated computational features; (2) cross-referencing between data sets and published articles; and (3) unifying collections of data sets, spanning multiple publications, into an integral whole. The central elements of **MOSAIC** (which stands for "Multiparadigm Ontologies for Scientific, Academic, and Technical Publishing") will be outlined below.

In response to the contemporary Covid-19 crisis, my company, Linguistic Technology Systems, proposes to develop a concrete implementation of the **MOSAIC** protocol targeting Covid-19 data sets, and particularly the "Covid-19 Open Research Dataset" (**CORD-19**) research collection recently published at the launch of a new White House initiative devoted to Covid-19. In particular, we propose to create a Standard Development Kit (**SDK**) that would facilitate the deployment of software targeting **CORD-19**. In order to explain the role of this proposed **SDK** in relation to the currently existing **CORD-19** resourses, this paper will examine the present state

of CORD-19, and will point out certain limitations of existing CORD-19 data that can be ameliorated with code libraries that would be bundled with our proposed CORD-19 SDK.

I Assessing the CORD-19 Collection

The sudden emergence of Covid-19 as a medical and governmental priority presents a unique case-study of the limitations of our existing research-data platforms. Publishers have, to some degree, recognized the extraordinary nature of the new coronavirus crisis and taken some commensurate measures; for example, several publishing houses (including Springer Nature, Wiley, Elsevier, the American Society for Microbiology, and the New England Journal of Medicine) have committed to releasing as open-access documents a collection of papers potentially helpful for doctors and policymakers responding to the pandemic — some newly published and some dating back several years (the older research concerns viruses biologically similar to Covid-19). On March 16, the White House announced a "call to action ... to develop new text and data mining techniques that can help the science community answer high-priority scientific questions related to COVID-19" and simultaneously released the CORD-19 resources, which had been curated by a consortium of industry and academic institutions. The CORD-19 collection (see [1]), described as a "dataset" by the White House, is actually a "machine-readable Coronavirus literature collection" which includes article metadata and (in most cases) publication text for over 13,000 coronavirus research papers. This resource is accompanied by links to open-access document collections hosted on portals such as Springer Nature and Wiley Online.

Over all, then, CORD-19 has two distinct parts: there are, first, links to web portals offering full access to a select group of publications (in human-readable formats like PDF and HTML) and, second, a much larger machine-readable corpus of article texts relevant to Covid-19. The CORD-19 collection was formulated with the explicit goal of promoting both text mining and data mining solutions to advance coronavirus research. This means that CORD-19 is intended to be used both as a document archive for text mining and as a repository for finding and obtaining coronavirus data for subsequent research. Unfortunately, CORD-19 is not particularly well structured for either text or data mining, for reasons that will be outlined shortly. To some extent, this reflects CORD-19 being conceived as just a starting point; a spur to further implementation. The White House announcement directly requests institutions to develop *further* technologies which would help scientists and jurisdictions to take advantage of CORD-19 as it was initially published. In short, CORD-19 was released with the explicit anticipation that industry or academia would augment the underlying data by layering on additional software. Our proposed CORD-19 SDK would do just that, as a component providing analytic capabilities which make the raw CORD-19 data more valuable, and also a toolkit through which other developers could create new solutions targeting CORD-19.

Lacunae in the raw CORD-19 data — the problems which require customized CORD-19 software — can be identified both in the text encoding through which CORD-19 functions as a corpus for text mining, and in the research data potentially made available via CORD-19 and its archived publications. These problems fall into three broad categories, which will be discussed on more detail over the next several paragraphs:

Transription Errors This problem arises from incorrect or simplified translation of publication texts into a machine-readable format, which can cause the machine-readable text archive to misrepresent the structure and content of documents, hindering text-mining technology that targets the archive.

Poorly Indexed Research Data Although CORD-19 provides a structured representation of a large collection of research *papers*, there is no easy-to-use tool for finding research *data* through CORD-19.

Poorly Integrated Research Data The research data which *can* be accessed through CORD-19 evinces a wide variety of technical fields and formats, with distinct software requirements; as a result, it is a difficult task to merge and integrate different data sets related to Covid-19. At present, CORD-19 does not include any software tools or computer code that would facilitate data integration.

With respect to text mining, an immediate problem lies in CORD-19's archive-construction methodology: especially, how the text was parsed from PDF files. This is a process which almost inevitably causes imprecise or inaccurate text representation, which can degrade the quality of the archive unless manual or automated corrections are made. In particular, the CORD-19 library evinces transcription errors, particularly in relation to technical or scientific phrases and terminology, and scientific notation may be improperly transcribed.

A formal designation such as "2'-C-ethynyl" (to take one specific case) is encoded in **CORD-19** as "2 0 -C-ethynyl", which could stymie text searches against the **CORD-19** corpus (see [2]). Moreover, there is no semantic marking identifying that the "2 0 -C-ethynyl" text segment has a specific technical meaning. These errors or limitations arise in part from unavoidable anomalies which occur when reading texts from **PDF** files rather than from machine-readable, structured formats such as **XML**.

To address these problems, our proposed **SDK** would try to cross-reference the **CORD-19 JSON** data against **XML** or LaTeX manuscripts which may have been saved by authors or publishers as part of the publication process. At its most complete, this would entail requesting authors whose papers are indexed on **CORD-19** to submit source materials if they are available in a machine-readable format, such as LaTeX, as well as requesting composit-related representations from affiliated publishers, with the goal of providing machine-readable access to publications in multiple formats. The **SDK** would then include built-in client libraries with procedures to merge different representations of each document.

It is also worth observing that the **JSON** format used for encoding **CORD-19** manuscripts presents some logistical challenges for any operations related to text-mining or to cross-referencing publications and data sets. In particular, **CORD-19** makes partial use of "standoff annotation"; specifically, document features such as citations and references are notated through character offsets into the paragraph where they appear. As a result, accurately reading these document elements requires synthesizing data points parsed from several distinct objects in the **JSON** code, which is only feasible given a client library built to interface with the **CORD-19** files in accord with their specific schema. Such a client library, again proposed as part of a **CORD-19 SDK**, would implement convenience procedures to handle recurring tasks, such as obtaining the full bibliographic reference affixed to a given location in a manuscript.

With respect to *data* mining in the **CORD-19** context, the limitations in the currently available raw **CORD-19** data are even more pronounced than in the context of text mining. In particular, neither the article metadata nor the full open-access document collections have any mechanism for actually obtaining data published alongside research papers, or even identifying which papers have accompanying data in the first place. The Springer Nature collection illustrates the limitations of this relatively unstructured data-publishing approach (this following analysis will focus on Springer Nature, but the problems identified are no less pronounced on the other **CORD-19** portals — if anything, because Springer Nature allows readers to browse articles in **HTML** within the web portal directly, one can ascertain whether research data exists for an article without downloading and reading it; with other **CORD-19** resources it is actually harder to locate supplemental data when available). As of mid-March, the Springer Nature portal encompassed 43 articles, of which 15 were accompanied by research data that could be separately downloaded (this number does not include papers that document research findings only indirectly, via tables or graphics printed inline with the text). Collectively these articles referenced over 30 distinct data sets (some papers were linked to multiple data sets), forming a data collection which could be a valuable resource for Covid-19 research — not only through the raw data made available but as a kernel around which new coronavirus data could accumulate. However, there is currently no mechanism to make this overall collection available as a single resource.

This problem demonstrates, among other things, how the Research Object protocol is limited in applying only to *single* articles. As a result, there is no commensurate protocol for publishing *groups* of articles which are tied to groups of data sets unified into an integral whole. Open-access Covid-19 papers also reveal limitations of exiting online document portals, particularly with respect to how publications are linked to data sets. In particular, there is no clear indication that a given paper is associated with downloaded data; usually readers ascertain this information only by reading or scrolling down to a "supplemental materials" or "data availability" addendum near the end of the article. Moreover, because the Springer Nature portal aggregates papers from multiple sources, there is no consistent pattern for locating data sets; each journal or publisher has their own mechanism for alerting readers to the existence of open-access data and allowing them to download the relevant data sets.

To address the **CORD-19** text and data mining limitations, our **SDK** would provide capabilities including (1) a client library for manipulating and examining the **CORD-19 JSON** files; (2) supplementing the **CORD-19** files with alternative document representations which would offer a more detailed infoset for document content, wherever such files are available; (3) a machine-readable index of open-access data sets linked to **CORD-19** articles; and (4) software tools for downloading and manipulating these coronavirus data sets. In addition, our **SDK** would provide tools for merging and integrating Covid-19 data from disparate sources. The problems of data integration, and solutions offered by the proposed **SDK**, are discussed in the following paragraphs.

II Data Integration within CORD-19

Aside from the issues, described over the last several paragraphs, which are likely to hinder text and data mining across **CORD-19**, the collective group of Covid-19 data sets also illustrate the limitations of information spaces pieced together from disconnected raw data files with little additional curation. The files included in this group of data sets encompass an array of file types and formats, including **FASTA** (which stands for Fast-All, a genomics format), **SRA** (Sequence Read Archive, for **DNA** sequencing), **PDB** (Protein Data Bank, representing the **3D** geometry of protein molecules), **MAP** (Electron Microscopy Map), **EPS** (Embedded Postscript), and **CSV** (comma-separated values). There are also tables represented in Microsoft Word or Excel formats. Although these various formats are reasonable for storing raw data, not all of them are actually machine-readable; in particular, the **EPS**, Word, and Excel files need manual processing in order to use the information they provide in a computational manner. A properly curated data collection would need to unify disparate sources into a common machine-readable representation (such as **XML**).

Going further, productive data curation should also aspire to *semantic* integration, unifying disparate sources into a common data model. For example, multiple spreadsheets among the Springer Nature Covid-19 data sets hold sociodemographic and epidemiological information relevant to modeling the spread of the disease. These different sources could certainly be integrated into a canonical social-epidemiology-based representational paradigm which recognizes the disparate data points which might be relevant for tracking the spread of Covid-19 (with the potential to unify data from many countries and jurisdictions). This is not only a matter of data *representation* (viz., how data is physically laid out), but also of data types and computer code. According to the Research Object protocol, data sets should include a code base which provides convenient computational access to the published data. In the case of Covid-19, this entails creating a sociodemographic and epidemiological code library optimized for Covid-19 information, which would be the primary access point for researchers seeking to use the data which has been published in conjunction with the 43 manuscripts examined here that were aggregated on Springer Nature, along with any other coronavirus research which comes online. Similar comments apply not only to tabular data represented in spreadsheet or **CSV** form, but to more complex molecular or microscopy data that needs specialized scientific software to be properly visualized.

Considering the overall space of Covid-19 data, it is unavoidable that some files require special applications and cannot be directly merged with the overall collection. For instance, there is no coherent semantics for unifying Protein Data Bank files with social-epidemiology; files of the former type have specific scientific uses and can only be understood by special-purpose software. Nevertheless, a well-curated data-set collection can make using such special-purpose data as convenient as possible. In the case of Protein Data Bank, a downloadable Covid-19 archive can include source code for **IQmol**, a molecular-visualization application that supports **PDB** (among other file formats) and has few external dependencies (so it is relatively easy to build from source).

Indeed, a curated Covid-19 archive might include an enhanced version of **IQmol** prioritizing Covid-19 research, with the goal of integrating biomolecular and social-epidemiological data as much as possible. For example, as Covid-19 potentially mutates in different ways in different geographic areas, it will be important to model the connections between "hard" scientific Covid-19 information and sociodemographics. As the pandemic evolves, genomic and biochemical information may be linked to particular strains of virus, which in turn are linked to sociodemographic profiles: certain strains will be more prevalent in certain populations. Consequently, models of Covid-19 variation will need to be formulated and then integrated with both chemical/molecular data and sociodemographic/epidemiological data. Different Covid-19 strains then form a bridge linking these different sorts of information; researchers should be able to pass back and forth from molecular or genomic visualizations of Covid-19 to social-epidemiological charts and tables based on viral strains. Ideally, capabilities for this sort of interdisciplinary data integration would be provide by a Covid-19 archive as enhancements to applications, such as **IQmol**, that scientists would use to study the published data.

It is worth noting that a data-mining platform requires *machine-readable* open-access research data, which is a more stringent requirement than simply publishing data alongside which can be understood by domain-specific software. For example, radiological imaging can be a source of Covid-19 data insofar as patterns of lung scarring, such as "ground-glass opacity", is a leading indicator of the disease. Consequently, diagnostic images of Covid-19 patients are a relevant kind of content for inclusion in a Covid-19 data set (see [7] as a case-study). However, diagnostic images are not in themselves "machine readable"; when medical imaging is used in a quantitative context (e.g. performing Machine Learning for pathology) it is nesessary to perform

Image Analysis to convert the raw data (viz., in this case. radiological graphics) into quantitative aggregates (for instance by using image segmentation to demarcate geometric boundaries and then defining diagnostically relevant features, such as opacity, as a scalar field over the segment). In short, even after research data is openly published by article authors, it may be necessary to perform aditional analysis on the data for it to be a full-fledged component of a machine-readable information space. Even within the limited group of data sets linked specifically to the Springer Nature articles, some of the published data actually comprises image graphics that need to be converted to numerical formats to work with **CORD-19** overall.

Another concern in developing an integrated **CORD-19** data collection is that, logistically speaking, not all Covid-19 data is practical to reuse as a downloadable package. This is especially true for genomics; several of the aforementioned 43 coronavirus papers included data published via online data banks capable of hosting data sets that are too large for an ordinary computer. In these situation scientists formulate queries or analytic scripts that are sent remotely to the online repositories, so that researchers access the actual published data only indirectly. Nevertheless, access to these data sets can still be curated as part of a Covid-19 package; in particular, computer code can be provided which automates the process of networking with remote genomics archives through the accession numbers and file-formats which those archives recognize. More generally, our proposed **SDK** would provide a toolkit for composing client libraries targeting remote data services — formulating queries and then parsing the results delivered in formats such as **XML** or **JSON** — as well as concrete libraries for remote data sources repeatedly used by **CORD-19** data sets.

As a final point on the topic of integrating disparate **CORD-19** research data, note that an overarching framework for indexing Covid-19 data sets would also facilitate the process of cross-referencing article text and research data. In particular, our proposed **SDK** would introduce a system of *microcitations* that apply to portions of manuscripts *as well as* data sets. In the publishing context, a microcitation is defined as a reference to a partially isolated fragment of a larger document, such as a table or figure illustration, or a sentence or paragraph defining a technical term, or (in mathematics) the statement/proof of a definition, axiom, or theorem. In data publishing, "data citations" are unique references to data sets in their entirety or to smaller parts of data sets. A data microcitation is then a fine-grained reference into a data set: for example, "the precise data records actually used in a study" (as defined by the Federation of Earth Science Information Partners; see [5]), one column in a spreadsheet, or one statistical parameter in a quantitative analysis.

The unique feature we propose for our **SDK** would be to combine the text-mining and data-mining notions of microcitation into a unified framework. In particular, text-based searches against the **CORD-19** corpus would try to find matches in the data sets indexed by our **SDK**. As a concrete example, a concept such as "expiratory flow" appears in **CORD-19** both as a table column in research data and as a medical concept discussed in research papers; a unified microcitation framework should therefore map *expiratory flow* as a keyphrase to both textual locations and data set parameters. Similarly, a concept such as *2'-C-ethynyl* (mentioned earlier in the context of transcription errors) should be identified both as a phrase in article texts and as a molecular component present within compounds whose scientific properties are investigated through **CORD-19** research data, so that a search for this concept can trigger both publication and data-set matches. Implementing this kind of unified search mechanism requires that data sets be *annotated* with techniques similar to those used for marking up Natural Language techniques; consequently, the proposed **SDK** would include a custom data-annotation library, to be employed in conjunction with operations to aggregate **CORD-19** data sets into a common representational format. In addition, the **SDK** would include a customized document viewer which embeds code to recognize and interact with microcitations in both manuscript and data-set environments.

III The MOSAIC Protocol

As mentioned at the top, the **MOSAIC** data-publishing model prioritizes the implementation of *self-contained* data sets. "Self-Contained" means that researchers who download a data set will not need to download *additional* software in order to examine and reuse the data set. One important goal of the Research Object protocol is to make research data available with a minimum of additional effort. Accordingly, Research Object should try to minimize external dependencies — in particular, should minimize extent to which users need to install software beyond what is provided by the Research Object Bundle itself. This "stand-alone" status can be achieved in various ways. One option is to organize Research Objects around widely used scientific computing platforms, such as R, Jupyter, or Matlab; another is to bundle dependencies into a virtual platform, using tools such as ReproZip. An alternative solution, more in keeping with the vision of truly *self-contained* Research Objects — the one which is embraced by **MOSAIC** and would be adopted for our proposed **CORD-19 SDK** — is to provide a kind of out-of-the-box computing platform contained entirely within the downloadable

code and data. In this context, code bases which can be distributed in pure source fashion — such as WhiteDB for a database engine or AngelScript (an embedded C++-based scripting language) for a scripting engine — are especially valuable.

Considering the inter-disciplinary nature of Covid-19 research, it is unavoidable that different scientists will need different sorts of specialized software to analyze the kinds of information relevant to their research. The computational techniques applicable to diagnosing coronavirus infection are scientifically very different from those used for genomic or epidemiological studies of the disease; it is impractical to expect pathologists to use the same software as bioinformaticians studying the pathogen, or for either to use the same software as virologists modeling the potential or observed spread of the disease. In short, even if scientists from disparate disciplines start with a common pool of raw data, they will need to analyze this data through a diverse set of supplemental computational tools, which will vary not only across disciplines but also in terms of the software and laboratory facilities available to different researchers through their institutions. In this sense it is impossible to unify all CORD-19 data into a *fully* self-contained information space.

Nevertheless, committing to "standalone" data publishing remains a valuable goal even in a context where published data sets will inevitably migrate to different digital ecosystems. Although scientists may use external digital tools *when necessary* to perform certain calculations, or to interface with laboratory equipment, we can hope to minimize the *degree of variation* across different domain-specific extensions to CORD-19. Ideally, that is, the version of CORD-19 (along with its supporting technology) found in a biomedical setting will be as similar as possible to that found in a biochemical context, or a health-policy contest. By providing a self-contained computing framework is delivered as an integral package accompanying CORD-19, a multi-disciplinary CORD-19 software package (which could be organized through our proposed SDK) would limit the degree of technological divergence that would naturally tend to arise as CORD-19 is leveraged by different scientific fields. In the absence of any initiative to limit this drift, CORD-19 could easily devolve into a federation of separate resources which have no interconnection apart from their nominal focus on Covid-19.

The vision of *standalone* and *self-contained* Research Object Bundles reveals new criteria that can be introduced as goals for data publication, analogous to FAIR. In particular, Research Objects should be (1) *self-contained* (with few or no external dependencies), (2) *transparent* (meaning that all computing operations should be implemented by source code within the bundle that can be examined as code files and within a debugging session), and (3) interactive (meaning that the bundle does not only include raw data but also software to interactively view and manipulate this data). Research Objects which embrace these priorities will try to provide data visualization, persistence, and analysis through GUI, database, and scripting engines that can be embedded as source code in the Research Object itself. In particular, self-contained bundles can be organized around what in MOSAIC is called a "triple-kernel" (or M3K, for MOSAIC Triple Kernel) architecture, referring to database, data visualization, and scripting engines. A good prototype for triple-kernel implementation would gravitate toward components that can be unified into a common code base: for example, WhiteDB for the database kernel (see [6]), AngelScript for the scripting kernel (see [3]), and QT (see [4]) for the data visualization kernel (and also for networking). With the (freely available) QT libraries, each of these components can be built via the QT build system and distributed in source code fashion; there is no need to system-level installs or any other external build or install operations.

A fully self-contained M3K bundle as just described will support several different forms of analytic operations and queries: assuming the WhiteDB/AngelScript/QT architecture, this would include queries against the WhiteDB database engine, queries executed by running Angel scripts, and queries against remote services via QT Networking (to support non-downloadable data). Hypergraph code and data representation can then support a unified "Transparent Hypergraph Query Language" (THQL) which provides a common mechanism for accessing these different varieties of queries. THQL is implemented via the notion of "hypergraph construction as intermediate representation": specifically, the logic to construct queries is understood as a manifestation of the logic to populate hypergraphs (meeting certain formal criteria). Consequently, a hypergraph-construction language can serve as a kind of bytecode for compiling database, scripting, and networking queries in a common framework; THQL provides an Intermediate Representation and Virtual Machine which dispatches queries to the respective M3K kernels (in the prototype case, to the WhiteDB, AngelScript, and QT engines respectively). A canonical WhiteDB/AngelScript/QT M3K framework can be facilitated by a Standard Development Kit, extending the AngelScript SDK with WhiteDB and QT integration.

Having described a M3K model for *individual* Research Objects, one can then consider the development of data set *collections* encompassing multiple research papers. A protocol for Research Object aggregation can be developed by extending a M3K SDK with domain-specific code appropriate for the shared topics and

subject-matter which unify the presented research work. In *Advances in Ubiquitous Computing*, for example, most of the chapter discussed either signal processing (particularly in the context of bioacoustics and 5G wireless networks) or Natural Language Processing, so a combined code base would include a mixture of signal-processing and **NLP** libraries. Analogously, a **CORD-19 SDK**, as proposed here, could bundle code related to different facets of Covid-19 research, from demographic and quantitative epidemiological modeling to networking with remote genomic data.

IV Conclusion

This paper has highlighted limitations of data sets published in conjunction with coronavirus articles made available as open-access resources on Springer Nature (and, by extension, **CORD-19**). The central point here is to argue for a distinct data-curation stage in the publication process, with data curators playing a role distinct from that of both authors and editors. Moreover, the discussion has hopefully highlighted problems with current data-sharing paradigms, even those such as the Research Object and **FAIR** initiatives which are explicitly devoted to improving how open-access data sets are published. **CORD-19** exposes several lacunae in the Research Object protocol, for example, which point to the need for a more detailed extension of this protocol. In particular, an enhanced protocol should encompass:

1. A canonical framework for archiving collections of data sets, not only single data sets (and not only groups of data sets published with a single research paper). For example, all data sets published alongside the 43 Springer Nature articles could be unified into a single collection.
2. A code base accompanying data-set collections designed to help research unify the information provided. Curating the overall collection would involve pooling disparate data into common representation, and implementing computer code which deserializes and processes the unified data accordingly. For instance, **CSV**, **EPS**, and Microsoft Word/Excel tables could be migrated to **XML**, **JSON**, or a more complex common format (Chapter 3 of *Advances in Ubiquitous Computing* presents the theoretical case for a "software-centric" representational format based on hypergraphs). Customized computer code could then be implemented specifically to parse and merge the information present in single data sets within the overall collection. This implementation would reciprocate the Research Object goal of unifying code and data, but again would operate at the level of an aggregate of research projects rather than a single Research Object.
3. A unified data-set collection should be self-contained as much as possible, and should be built around a foundation where advanced computing capabilities are available in a transparent, standalone fashion, without requiring tools outside the collection itself. One way to achieve this is via a M3K architecture as outlined earlier; it is straightforward to publish the WhiteDB and AngelScpipt code bases within a Research Object collection, and to employ the **Qt** ecosystem (e.g., the **Qt** Creator Integrated Development Environment) as the underlying operational milieu for using and obtaining the data sets. For example, it would be possible to implement **Qt**-specific libraries for interfacing with the **CORD-19** library and with other Covid-19 dashboards and data sets.
4. A unified data-set collection should also provide prototyping and remote-access tools to interface with web-based information spaces that host data sets too large to be individually downloaded. Ideally, these would include simulations of remote services analogous to PurpleData vis-à-vis BigData, which would help scientists understand the design of the remote archives and how to interface with them.
5. Finally, a unified research portal should influence the design of the web portals where associated texts are published. It should be easy for readers to identify which articles have supplemental data files and to download those files if desired. Moreover, textual links should be established between publication content and data sets — for instance, a plot or diagram illustrating statistical or equational distributions should link to the portion of the data set from which that quantitative data is derived.

The above discussion has considered the Springer Nature articles as a case-study, but analogous comments would apply to other Covid-19 related resources. For example, John Hopkins University has created and deployed a Covid-19 "dashboard" tracking the spread of the virus (this is one of several dashboards that have come online for visualizing the evolution of the pandemic, with varying degrees of complexity); new data from which the web dashboard is generated is published via a **GIT** archive roughly once daily. If and when the reported Verily portal comes online, hopefully machine-readable access to that public data will be provided either via an analogous updated archive or via an **API**. Ideally, these disparate Covid-19 projects will be interoperable: any code published in relation to the Springer Nature coronavirus collection, for example, could include components implemented to access the John Hopkins and (anticipated) Verily data sets as well as all the data brought in via the **CORD-19** articles. In particular, a useful starting-point for Covid-19 data

integration would be a single data type (e.g., a C++ class) whose specific purpose is to obtain the most recent available coronavirus data: one procedure to download the latest files from the John Hopkins dashboard, one procedure to search Springer Nature for new relevant content, etc. Such a data type would then serve as a guide for obtaining Covid-19 information. Insofar as conscious effort is made to integrate all publicly accessible Covid-19 data via an overarching toolkit, it will be easier to continually accumulate new data sources as these come online.

This discussion has also used the Covid-19 crisis as a lens through which to examine data-publishing limitations in general. These problems are not specific to coronavirus, but the almost unprecedented urgency of this epidemic exposes how science and the publishing industry are still struggling to develop technologies and practices which keep pace with the intersecting needs of systematic research and public policy. An optimistic projection is that the crisis will spur momentum toward a more sophisticated data-sharing paradigm — perhaps a generalization of the Research Object protocol toward data-set collections, with features as outlined above. We hope to contribute to the emergence of such a protocol, so as to operationalize some of the ideas laid out in *Advances in Ubiquitous Computing*. It would be especially rewarding if an integrated data-set collection devoted to Covid-19 would serve as a first example and a test-bed for this new paradigm, given the potential public benefit of unifying disparate Covid-19 data as effectively as possible, where this technology can then be generalized to other medical priorities and other academic disciplines overall.

Supplemental to this overview, we can provide information about a proposed portal for publishing research papers and data sets conformant to the above enhancements to the Research Object protocol. In particular, we can share the following: (1) description and demonstrations of a Qt and hypergraph-based extension to WhiteDB, which we term WhiteCharmDB, that can be used to merge individual data sets and prototype remote-analytic interfacing logic; (2) documentation of a plugin mechanism, using IQmol as a case-study, which integrates document viewers with scientific applications and which would be especially useful for interconnecting publications and data sets in an integrated research portal; (3) illustrations of VersatileUX, a library of GUI components that could be adapted to provide custom front-ends for raw data published through an integrated research portal; and (4) an overview of an architecture for research portals which we call MOSAIC, that could guide the implementation of portals supporting data-set collections, such as a Covid-19 archive that could serve as a model for future work as well as a useful resource while the pandemic continues to present a global crisis.

References

[1] "COVID-19 Open Research Dataset (CORD-19)". 2020. Version 2020-03-13. Retrieved from https://pages.semanticscholar.org/coronavirus-research. Accessed 2020-03-20. doi:10.5281/zenodo.3715506 https://pages.semanticscholar.org/coronavirus-research

[2] Ludëk Eyer, *et. al.*, "Nucleoside analogs as a rich source of antiviral agents active against arthropod-borne flaviviruses". https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5890575/

[3] Andreas Jönsson, "AngelCode Scripting Library", www.AngelCode.com/AngelScript/

[4] Ivan Čukić, "Functional and imperative reactive programming based on a generalization of the continuation monad in the C++ programming language", http://elibrary.matf.bg.ac.rs/bitstream/handle/123456789/4738/ivan_cukic_phd.pdf?sequence=1

[5] Mark A. Parsons and Ruth Duerr, "Data Identifiers, Versioning, and Micro-citation", https://www.thelancet.com/action/showPdf?pii=S1473-3099%2820%2930086-4

[6] Enar Reilent, "Whiteboard Architecture for the Multi-agent Sensor Systems", https://www.thelancet.com/action/showPdf?pii=S1473-3099%2820%2930086-4

[7] Heshui Shi, *et. al.*, "Radiological findings from 81 patients with COVID-19 pneumonia in Wuhan, China: a descriptive study". https://www.thelancet.com/action/showPdf?pii=S1473-3099%2820%2930086-4