Proposing a CORD-19 SDK Solution to Improve Machine Readability

to coordinate with
SPRINGER NATURE, WILEY, ELSEVIER,
the AMERICAN SOCIETY FOR MICROBIOLOGY,
and the NEW ENGLAND JOURNAL OF MEDICINE

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.

One goal of *Advances in Ubiquitous Computing* is to advocate for a refined, more rigorous data publishing protocol, which is especially important at the current moment when the urgency of sharing Covid-19 research is boosting the contemporary open-access and data-centric publishing model. 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 introduce a new protocol for publishing research data, which essentially extends existing initiatives such as Research Objects and **FAIR** (which stands for "Findable, Accessible, Interoperable, Reusable"). In principle, these initiatives — which have been pushed by publishers, academic institutions, and certain government agencies (such as the National Institutes of Health) — are intended to make it easier for scientists to find, assess, reuse, and re-produce research data. In practice, however — due partly to technological limitations and partly to authors and publishers being slow to adopt the new protocols — the ecosystem of published data sets remains far less rigorously organized than the ecosystem of published scientific/academic books and articles.

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 extra-ordinary 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 a "Covid-19 Open Research Dataset" (CORD-19) that was curated by a consortium of industry and academic institutions**. This "dataset" 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.

Even the goal of opening these materials for access to the general public is, at least in part, to spur *data mining* for Covid-19 research, the is that all of the resources aggregated via CORD-19 are poorly designed for this priority. The reason why is because 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 — which had already come online some time before **CORD-19** was officially announced — illustrates the limitations of this relatively unstructured data-publishing approach. This paper will Springer Nature allows readers to browse articles in **HTML** within their web portal; with other **CORD-19** resources it is actually harder to locate supplemental data, because in general there is no way to ascertain whether research data exists for an article without downloading and reading it). 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. 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. 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.

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 could implement convenience procedures to handle recurring tasks, such as obtaining the full bibliographic reference affixed to a given location in a manuscript.

A further complication in the **CORD-19** arises from how the text was parsed from **PDF** files, a process which can cause imprecise or inaccurate text representation. For example, scientific notation may be improperly transcribed; a formal notation 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. 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, it would be optimal 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. Here, too, a **CORD-19 SDK** could include built-in client libraries with procedures to merge different representations of each document. As well as facilitating text-mining across the corpus, these client libraries would also improve the usability of **CORD-19** data sets; in particular, it would be possible to search the document corpus for terms which appear in data sets, thereby building a network of links between data sets and publications.

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 inluded 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 instead, as much as possible, unify disparate sources into a common machine-readable representation (such as **XML**).

Going further, productive data curation would 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.

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 for-

mulate 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. A case-study in this type of technology is provided by Verily, the Alphabet division which (according to news reports, such as a New York Times cover-page story on March 15th) is developing an online platform for the public to check symptoms and locate providers prepared to treat Covid-19 cases. One Verily project is PurpleData, esssentially a miniature version of Google's "BigData" platform; unlike its larger archetype, PurpleData is designed to run and host data sets small enough for a single computer, so as to develop and test analytic queries which are eventually posted to remote BigData services. In short, PurpleData is a simulation of BigData designed for prototyping code which will interface with BigData. This prototyping/simulation paradigm is a useful model to apply to other remote-analytics services, such as the National Center for Biotechnology Information (NCBI) genomics archives where some of emerging Covid-19 data has been stored. The point here is not to recommend a direct generalization of PurpleData — indeed, PurpleData (implemented in Python) is not an ideal case-study in prototyping/simulation techniques because it is not legitimately self-contained. One could argue that alternative technologies, such as WhiteDB — an under-appreciated database engine (an SQL/NoSQL hybrid) currently used for certain CyberPhysical and telemedical systems — would be a more definitive framework for prototyping solutions. WhiteDB is a standalone C/C++ engine which can be dropped as source code into any compiled project; it therefore offers fully transparent access to data-serialization code, and would be especially useful for emulating the structural and persistence methodologies implemented at large scale by cloud-analytic services such as BigData or NCBI Nucleotide/GenBank.

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, arguably more in keeping with the vision of truly *self-contained* Research Objects, 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.

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 might be called a "triple-kernel" (or T3K, for transparent-triple-kernel) architecture, referring to database, data visualization, and scripting engines. A good prototype for triple-kernel implementation would use WhiteDB for the database kernel, AngelScript for the scripting kernel, and QT 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 T3K 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 Whit-

eDB 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 T3K kernels (in the prototype case, to the WhiteDB, AngelScript, and **Qt** engines respectively). A canonical WhiteDB/AngelScript/**Qt** T3K framework can be facilitated by a Standard Development Kit, extending the AngelScript **SDK** with WhiteDB and **Qt** integration.

Having described a T3K 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 T3K **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** could bundle code related to different facets of Covid-19 research, from demographic and quantitative epidemiological modeling to networking with remote genomic data.

This discussion 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 point here is not to criticize the work of individual authors, but rather 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 T3K 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.