

Executable Music Documents

David De Roure
University of Oxford
Oxford e-Research Centre
Oxford OX1 3QG, UK
+44 (0)1865 610703
david.deroure@oerc.ox.ac.uk

ABSTRACT

While good practices are emerging with respect to publication of data alongside research outputs, we argue that computational descriptions (e.g. scripts, software and workflows) should also be included so that research can be interpreted, reconstructed and recomputed. A research article—or *Research Object*—should then describe all the components associated with a piece of digital research, including the descriptions of code and algorithms, effectively comprising an executable document. Furthermore we observe that such a re-executable object can be re-run automatically. The Music Information Retrieval research community has established community infrastructure and practices which are amenable to this approach, providing a glimpse of a future Music Digital Library. These ideas raise a number of issues for Digital Libraries more generally.

Categories and Subject Descriptors

H.3.7 [Information Storage and Retrieval]: Digital Libraries – collection.

General Terms

Documentation.

Keywords

Digital Music Object, Executable paper, Literate Programming, Reproducibility, Research Object, Software Sustainability.

1. INTRODUCTION

This paper discusses the reusable research record that we need for digital scholarship today and tomorrow. It draws together three lines of thinking and experience into an extended notion of executable document, exemplified here in the music domain. The three lines of thinking are literate programming, software publication, and research objects. These represent three different but related ambitions which we believe necessarily come together in establishing the reusable record for digital research. This in turn has significant implications for libraries, curation and publishing, as we consider increasingly automated practice in digital scholarship.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from Permissions@acm.org.

DLfM '14, September 12 2014, London, United Kingdom
Copyright is held by the owner/author(s). Publication rights licensed to ACM.

ACM 978-1-4503-3002-2/14/09...\$15.00.
<http://dx.doi.org/10.1145/2660168.2660183>

The signal processing community has been close to these issues for a long time, with the “WaveLab and Reproducible Research” report published in 1995 [1], and the audio and music community is excellently positioned as an early adopter of new approaches through its end-to-end adoption of digital techniques. We suggest this is a powerful case study.

Section 2 introduces the three lines of thinking, Section 3 discusses executable documents and applies the notion to the music information retrieval domain as a symbiotic case study, and the paper concludes by making observations in Section 4 to inform the future digital libraries research agenda.

2. BACKGROUND

2.1 Literate Programming

Donald Knuth introduced the notion of “Literate Programming” in 1984:

“I believe that the time is ripe for significantly better documentation of programs, and that we can best achieve this by considering programs to be works of literature... Instead of imagining that our main task is to instruct a computer what to do, let us concentrate rather on explaining to human beings what we want a computer to do.” [2]

This was facilitated by the provision of tools, and there continue to be successful systems (such as *sweave* [3]) in development and use today. As well as being descriptive, the term literate programming was chosen to impose a moral commitment by discouraging “illiterate programming”. It makes for understandable and reproducible code which produces results that can be properly interpreted. It is an important part of our story because it makes it easier for *humans* to interpret, recreate and re-use research. Indeed *in extremis* “programs must be written for people to read, and only incidentally for machines to execute” [4]. One might argue that a well explained algorithm is better for reproducibility than the code alone, because it facilitates independent re-implementation.

2.2 Software Publication

There is a flourishing software community and industry with excellent practices in community software development and its sustainability¹. In the music domain, the Sound Software institute², based at the Centre for Digital Music at Queen Mary University of London, focuses in particular on ‘sound software’ for audio and music research [5], reflecting the sophistication in this domain. Those concerned with the future of scholarly communication and data sharing stand to gain many insights from the software development community; e.g. releases, versioning, collaborative authoring, automated building and testing.

¹ Software Sustainability Institute, <http://software.ac.uk>

² Sound Software, <http://soundsoftware.ac.uk>

Meanwhile there are efforts to fit software (source code) into established publishing models—so that others might use and cite it, so that the research is more reproducible and, significantly, so that the authors might get credit for it. Hence there are various efforts by publishers to publish software, notably the *Journal of Open Research Software* from Ubiquity Press³ and the short-lived *Open Research Computation* from BioMed Central. Some publishers also embrace software as supplementary material. Yet it is not a common practice and software as the subject or supplement of a publication has failed to attract the same attention as data. Arguably it is at least as important [6].

Software is machine-processable. In pursuit of curated archives and replicable research there might be an argument for preserving executable software, perhaps running old code through the use of virtual machines to emulate historic hardware platforms. However, treating software as a “black box” is problematic for interpretation and reproducibility.

2.3 Research Objects

Broadly the notion of the Research Object is that a researcher can bundle together all the bits and pieces that make up the record of a piece of research into one sharable and citable object; i.e. the evidence for a research outcome or a decision [7]. By aggregating the multiple digital pieces into one object with one identifier we achieve a new sharable, citable social object which drops into the tooling of digital research. Crucially the content can be distributed and its components might be exchanged with computers as well as humans. Research Objects need not contain software or executable parts, but typically might as records of digital research.

An early example of a Research Object is the *pack* in myExperiment⁴, a social website for sharing computational workflows [8]. Conceived with workflows as the ‘social object’, by analogy with photos on flickr and movies on YouTube, myExperiment users soon requested the ability to attach data, logs, papers, presentations etc to their workflows. This led to the notion of packs essentially as bundles of URLs pointing at the distributed content. myExperiment packs are represented using the Open Archives Initiative Object Reuse and Exchange (OAI-ORE) standard and available as linked data, hence semantically-described for ease of discovery and reuse. myExperiment’s particular notion of workflow-centric Research Objects has been much more fully developed in the Wf4Ever project [9].

Our analysis of myExperiment packs [10] led to a reflection on the nature and purpose of Research Objects, known informally as “the R Dimensions” [11]. This teases apart various aspects of the role of Research Objects. One important aspect is that they should be reusable, another is repurposeable (e.g. self-describing). More generally they enable people or machines to *reconstruct* a piece of research. This helps with reproducibility, but note that a Research Object is not “reproducible research” by itself: reproducibility means reusing a Research Object with a change to some circumstances, inputs, resources or components in order to see if the same results are achieved independent of those changes. Hence Research Objects should be amenable to such tests.

Today workflow-centric Research Objects are typically shared by humans and executed by hand, e.g. using the appropriate workflow workbench. When new data is available people might choose to rerun experiments to achieve new results, generating

new Research Objects. However there is no real need for humans to press the button, as the objects can be executed automatically. We see some of this today in the automatic execution of workflows to validate, maintain and even repair them.

Hence we anticipate the evolution of the notion of Research Object into the *Computational Research Object* (CRO), a model that enables machines to assemble and execute systems of Research Objects. This is essentially a computational model for Research Objects, indeed in generality it is a distributed computational model, and enables us to apply computer science approaches to the issues of future reproducible research.

CROs describe process (method) for machine enactment/execution and the associated digital resources, and are defined to be [12]: (1) *Social Objects* designed to facilitate human interpretation (e.g. containing narratives) and shared as part of a (hybrid) sensemaking network; (2) *Machine Objects* semantically described and programmatically accessible, designed for automation, scale and heterogeneity; (3) *Composable* with a distributed computational model, so that a Computational Research Object can itself assemble systems of objects, and these systems may consume and produce Computational Research Objects (such a system can be a subject of formal validation).

3. EXECUTABLE MUSIC DOCUMENTS

The idea of executable documents has been explored in the community; e.g. Elsevier’s *Executable Paper Grand Challenge* in 2011⁵. Here we revisit it by bringing together these three lines of experience and thinking; i.e. papers enhanced with computational descriptions (for human use), software publication, and new research records with computational components. In all cases we have citable, shareable objects for interpretation (or execution) by human and machine. Our view is that *machines are users too* and that these executable social objects are communicated in a sense-making network comprising both humans and machines.

To exemplify this, consider the *Executable Thesis*. A student includes their software in their (digital) thesis document, such that the thesis can be re-computed. So, for example, when new data comes along the thesis can be rerun and produce new results. The thought of the thesis becoming an automated paper factory is in some ways compelling. This thinking applies similarly to the executable article, journal or monograph.

Imagine now that the thesis gets executed automatically when the new data appears—and the results that arise could in turn prompt re-execution of other research. This might sound like science fiction but is only a slight extension of what we see happening today in the conduct of digital research; arguably it is a necessary extension if today’s research is to be properly recorded and reused, and to provide a foundation for digital innovation at scale.

Music Information Retrieval research is an excellent case study. The individual researcher might be running algorithms over a corpus of music, and as time goes on there is new input data (for example with the uploading of live music performances online). These computations may be long-running, and might also involve machine learning. So an experiment is not necessarily a one-off short lived and documented exercise, rather the researcher sets up a system (like a circuit) and monitors (or is notified of) new results. There is considerable automation in the research, but crucial human inspection and interpretation also. How is it to be recorded and re-used?

³ <http://openresearchsoftware.metajnl.com>

⁴ myExperiment, <http://www.myexperiment.org>

⁵ Executable Papers, <http://www.executablepapers.com>

This challenge scales to the broader Music Information Retrieval community. In the Music Information Retrieval Evaluation Exchange (MIREX) [13] the international community comes together annually to run improved algorithms and share the outcomes. This coproduction of research has led to an automated infrastructure for remote execution, and practices for annotation. Algorithms are reported as papers in the ISMIR conference and the community has practices for sharing code and data.

4. RESEARCH AGENDA

We suggest that MIREX can be regarded as an early Music Digital Library which has embraced software, execution and annotation, in a community-oriented approach to reusable, reconstructable and reproducible digital scholarship. As such it provides an excellent case study for taking a workflow and Research Object perspective [14] and we suggest this can extend to Executable Music Documents. At the same time it presents a study of scholarship, and the scholarly record, in a domain which is largely digital and stands to inform other areas.

This symbiosis can be taken further. We observe that the music industry itself is now essentially a digital system from end to (almost) end: from composition through to performance, recording, production, distribution and consumption, adopting techniques which include Semantic Audio and Linked Data. This itself might provide some lessons for scholarly communication. Furthermore, we can map the idea of a Research Object to a Digital Music Object which enables ease of reuse and remixing of music right through the chain from composition to consumption, with the consumer equally empowered to produce and compose.

The executable documents we present here go beyond the current discussions in the reproducible research community, as we glimpse an automated world of document execution where the digital library content gets on with the research in conjunction with humans. It raises a set of questions for Music Digital Libraries but also Digital Libraries generally:

1. Will digital libraries provide the infrastructure to execute documents, or will people deploy them on alternative infrastructures? What are the implications for discovery, curation, and its automation?
2. Who gains credit and owns the intellectual property generated when a document runs automatically? Who is liable for damage that arises? What are the implications of unintended or accidental assembly of research methods and outcomes?
3. What are the implications of research that occurs at very high speed, possibly speculatively, without human intervention? Where is the (critical, creative, subversive) human in the loop? Are we 'burning' research methods into an automated research platform?
4. How do executable documents sit in the social websites of discovery, authoring, publishing and sharing; i.e. the ecosystem of *scholarly social machines*?

5. ACKNOWLEDGMENTS

The author's work has been supported by Transforming Musicology (AHRC) and the Software Sustainability Institute (EPSRC EP/H043160/1). Thanks to Sean Bechhofer, Iain Buchan, Ichiro Fujinaga, Carole Goble, Brian Hole, Peter Murray-Rust, Cameron Neylon, Kevin Page, Mark Plumbley, Carolin Rindfleisch, Sweitze Roffel, Mark Sandler, David Shotton, Ségolène Tarte, Anita de Waard, Geraint Wiggins, Pip Willcox and Jun Zhao for discussions of ideas in this paper.

6. REFERENCES

- [1] B. Buckheit and D. L. Donoho. 1995. WaveLab and Reproducible Research, Dept. of Statistics, Stanford University, Tech. Rep. 474.
- [2] D.E. Knuth. 1984. Literate Programming. The Computer Journal. 27 (2): 97–111. British Computer Society.
- [3] F. Leisch. 2002. Sweave: Dynamic generation of statistical reports using literate data analysis. In W. Härdle and B. Rönz (editors), *Compstat 2002 - Proceedings in Computational Statistics*, pages 575-580. Physica Verlag, Heidelberg.
- [4] A.J. Perlis. 1985. Foreword to *Structure and Interpretation of Computer Programs* by H. Abelson and G. Jay Sussman with J. Sussman. MIT Press (1st ed.).
- [5] C. Cannam, L. A. Figueira and M. D. Plumbley. 2012. Sound software: Towards software reuse in audio and music research. In: Proc IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2012), Kyoto, Japan, pp 2745-2748, 25-30.
- [6] D. De Roure and C. Goble. 2010. Anchors in Shifting Sand: the Primacy of Method in the Web of Data. In Proceedings of WebSci10, April 2010, Raleigh, North Carolina, US.
- [7] S. Bechhofer, I. Buchan, D. De Roure, P. Missier *et al.* 2013. Why Linked Data is Not Enough for Scientists, Future Generation Computer Systems, Vol. 29, No. 2. pp. 599-611.
- [8] D. De Roure, C. Goble and R. Stevens. 2009. The design and realisation of the myExperiment Virtual Research Environment for social sharing of workflows. Future Generation Computer Systems 25 (5): 561–567.
- [9] K. Belhajjame, O. Corcho, D. Garijo, J. Zhao *et al.* 2012. Workflow-Centric Research Objects: A First Class Citizen in the Scholarly Discourse. In ESWC2012 Workshop on the Future of Scholarly Communication in the Semantic Web (SePublica 2012), Heraklion, Greece, May 2012.
- [10] D. De Roure, S. Bechhofer, C. Goble and D. Newman. 2011. Scientific Social Objects: The Social Objects and Multidimensional Network of the myExperiment Website. In 1st International Workshop on Social Object Networks (SocialObjects 2011), Boston, MA, US.
- [11] D. De Roure. 2010. Replacing the Paper: The Twelve Rs of the e-Research Record. e-Research, Nature blogs. <http://www.scilogsg.com/eresearch/replacing-the-paper-the-twelve-rs-of-the-e-research-record/>
- [12] D. De Roure. 2013. Towards computational research objects. In Proceedings of the 1st International Workshop on Digital Preservation of Research Methods and Artefacts (DPRMA '13) co-located at Joint Conference on Digital Libraries 2013, Indianapolis, IN, USA.
- [13] J.S. Downie, A.F. Ehmann, M. Bay and M.C. Jones. 2010. The Music Information Retrieval Evaluation eXchange: Some Observations and Insights. *Advances in Music Information Retrieval* Vol. 274, pp. 93-115.
- [14] K.R. Page, B. Fields, D. De Roure, T. Crawford, J.S. Downie. 2013. Capturing the workflows of music information retrieval for repeatability and reuse. *J. Intell. Inf. Syst.* 41(3): 435-459.