Article:
Hodge, Victoria Jane orcid.org/0000-0002-2469-0224, Jessop, Mark David, Fletcher, Martyn Anthony et al. (6 more authors) (2016) A Digital Repository and Execution Platform for Interactive Scholarly Publications in Neuroscience. Neuroinformatics. pp. 23-40. ISSN 1559-0089

https://doi.org/10.1007/s12021-015-9276-3

A Digital Repository and Execution Platform for Interactive Scholarly Publications in Neuroscience

Victoria Hodge[a], Mark Jessop[a], Martyn Fletcher[a], Michael Weeks[a], Aaron Turner[a], Tom Jackson[a], Colin Ingram[b], Leslie Smith[c] & Jim Austin[a]

[a]*Department of Computer Science, University of York, York, UK*
[b]*Institute of Neuroscience, Newcastle University, Newcastle upon Tyne, UK*
[c]*Dept of Computing Science and Mathematics, University of Stirling, Stirling, UK*

Corresponding Author:

Dr Victoria J. Hodge.
Dept of Computer Science,
University of York,
Deramore Lane,
York, UK
YO10 5GH
Email: victoria.hodge@york.ac.uk
Phone: +44 (0)1904 325637
Fax: +44 (0)1904 325599

Abstract:
The CARMEN Virtual Laboratory (VL) is a cloud-based platform which allows neuroscientists to store, share, develop, execute, reproduce and publicise their work. This paper describes new functionality in the CARMEN VL: an interactive publications repository. This new facility allows users to link data and software to publications. This enables other users to examine data and software associated with the publication and execute the associated software within the VL using the same data as the authors used in the publication. The cloud-based architecture and SaaS (Software as a Service) framework allows vast data sets to be uploaded and analysed using software services. Thus, this new interactive publications facility allows others to build on research results through reuse. This aligns with recent developments by funding agencies, institutions, and publishers with a move to open access research. Open access provides reproducibility and verification of research resources and results. Publications and their associated data and software will be assured of long-term preservation and curation in the repository. Further, analysing research data and the evaluations described in publications frequently requires a number of execution stages many of which are iterative. The VL provides a scientific workflow environment to combine software services into a processing tree. These workflows can also be associated with publications and executed by users. The VL also provides a secure environment where users can decide the access rights for each resource to ensure copyright and privacy restrictions are met.

Keywords.
Interactive publications; collaboration; reproducible neuroscience; interactive publication repository; execution platform; web portal; scientific workflows.

The final publication is available at Springer via http://dx.doi.org/10.1007/s12021-015-9276-3

1 Introduction.

Many funding agencies, institutions, and publishers have policies regarding open access to research data, code and software related to publications as transparency, openness and reusability are considered to be vital for scientific progress (Borgman, 2012; Lakhani, 2007). This implies integrated data, software and publication preservation and curation which have only very recently been explored. Creating research data and software applications and routines is a complex and expensive process. At the end of projects, much data and software are simply archived locally rendering it invisible to the wider research community. This is a particular problem where the data or software has been used in a publication. As a result, other researchers cannot cite or analyse this data themselves and scientific results cannot be validated. Borgman (Borgman, 2012) gives four rationales for preserving data:

1. reproducibility and verification;
2. open access to publicly funded research;
3. allowing interrogation of data; and,
4. advancing the state of the art in research.

The lack of visibility and access to research output can lead to repetition in research effort (for example, many fundamental software functions are routinely rewritten in labs) and inhibits the capability for others to build upon the results.

The CARMEN Virtual Laboratory (VL) forms part of the Code Analysis, Repository, and Modelling for e-Neuroscience (CARMEN) project. Our vision is a collaborative neuroscience VL for researchers, professional societies, publishers, editors and libraries which allows research outputs to be captured and made available for reuse. This paper describes a new facility that is now available: CARMEN VL Interactive Publications. Austin et al. (2011) proposed a publications facility which has now been developed and is detailed here.

The VL has an interactive publications repository where users link data, software services and workflows to documents (publication records). The data and its associated metadata can be explored by other users and the software and workflows can be explored and run in situ. The documents can be hosted on any website as a link can be created to an external URL, for example on a publisher's website or other publication collection. The VL allows users to capture and redistribute the outputs from neuroscience publications with particular emphasis on capturing the data and software outputs, creating scientific workflows and providing long-term interactive access to the data and software described in publications (Austin et al., 2011). Wider access to research results from electronic publications has both expedited and improved the effectiveness of follow-on research in many scholarly disciplines (Choudhury, 2008). In the VL, users upload their own data and software applications and create scientific workflows which they can now link to their publications to be assured of long-term preservation and curation. The VL is interactive and allows other users to execute the software and workflows, in situ, thus processing and analysing the data used in the publication. They can also analyse other data sets available in the VL. It is underpinned by a heterogeneous distributed compute platform which allows large data sets to be uploaded and analysed. Users have successfully uploaded data files up to 128GB in size into the system's storage. Interactive publications are realised through the use of the VL's Cloud and SaaS (Software as a Service) framework.

The following sections review similar systems and then describe the CARMEN VL interactive publications repository in more detail: the VL infrastructure; how the VL meets users' objectives; creating interactive publications including associating data, services and

workflows – **aimed at authors**; searching and exploring publications – **aimed at all users**; an overview of the CARMEN VL data and services; and, conclusions and future work.

2 Background and Overview

There are a number of general purpose on-line scientific archival environments: FigShare (Singh, 2011), DataONE (Michener et al., 2011) and emerging archiving facilities: DataCite (Brase, 2009), DataDryad (DataDryad, 2013). Scientific archival environments such as DataCite, DataDryad and FigShare aim to allow users to store data and metadata. They also provide users with persistent URLs to the datasets known as Digital Object Identifiers (DOI, 2013) (DOIs). Data can then be shared and cited through the DOIs.

A number of neuroscience platforms aim to be collaborative and act as a repository for data, code, publications or information. Neuroscience Information Framework (NIF) is a dynamic repository of Web-based neuroscience resources: data, a neuroscience ontology, materials, and tools (Gardner, et al, 2008). Neuroimaging Informatics Tools and Resources Clearinghouse (NITRC) users can find and compare neuroimaging tools for functional and structural neuroimaging analyses (Luo, Kennedy & Cohen, 2009). The NITRC portal is now moving towards virtual computing and data storage. The International Neuroinformatics Coordinating Facility (INCF) provides two platforms: Dataspace (INCFData, 2013) and Software Center (Ritz, et al., 2008). Dataspace allows users to find and share neuroscience data while the Software Center allows users to find, research and download software tools. The Collaborative Research in Computational Neuroscience (CRCNS) website provides for the sharing of tools and data including physiological recordings from sensory and memory systems and eye movement data (Teeters et al., 2008). Similarly, the German Neuroinformatics Node (G-Node) is a web portal for data access, data storage and data annotation focusing on cellular and systems neurophysiology data (Herz, et al., 2008). NeuroMorpho.Org (Ascoli, Donohue & Halavi, 2007) is a large repository of digital neuromorphological reconstructions. The reconstructed neurons can be hyperlinked to the publications describing them and are available for use in different research projects. BrainInfo (http://braininfo.org) is a growing repository of links to neuroscientific documents on the Web. It is indexed by NeuroNames (Bowden, et al, 2012), which is an ontology designed to handle ambiguities in neuroanatomical nomenclature. The Child and Adolescent NeuroDevelopment Initiative have developed CANDIShare, a portal for sharing structural brain images along with their anatomic segmentations and demographic data (Kennedy, et al., 2012). The MEG-SIM neuroscience portal (Aine, et al., 2012) provides access to an extensive test bed of realistic simulated MEG data and results of MEG data analyses across visual, auditory, and somatosensory modalities.

Other systems are aimed at general academic collaboration such as ResearchGate (www.researchgate.net) and the Sakai Project (http://www.sakaiproject.org). These allow users to share publications and data and provide social networking features but do not provide an execution environment. Research Objects proposed by Bechhofer et al. (2013) are containers which aggregate essential information relating to scientific experiments including data and metadata describing the methods used to produce and analyse the data and the people involved in the investigation. They can be used in conjunction with the myExperiment website (Goble et al., 2010). myExperiment allows users to upload their workflows and share them with other users or link to them in publications. However, the workflows cannot be viewed, edited or executed.

These collaborative repositories are data and reference resources and are complementary to the CARMEN VL. They allow users to share data and some use metadata or ontologies to add valuable information to data and other materials but they do not allow software and scientific workflows to be executed online which is the purpose of the CARMEN VL.

Other portals allow users to execute code online but generally do not allow users to upload and run their own code except in limited domains or using limited programming languages. Süptitz, Weis and Eymann (2013) provide a comprehensive literature review (in German) of such online resources. The Neuroscience Gateway (NSG) (Sivagnanam et al., 2013) is a portal for computational neuroscience providing neuronal simulation tools for parallel execution on complex high performance computing machines. The Portal allows users to run parallel neural simulations using existing modelling tools provided by NSG. However, users cannot upload and run their own software applications. Other platforms are aimed at specific domains. GenePattern (Reich et al., 2006) is a web-based system of tools that allows the creation of workflows for reproducible *in silico* research in genomics. Software tools can be composed into complex workflows and the system provides support tools such as data visualizers. Similarly, Mobyle, (Néron, et al. 2009) is a web-based workflow system for defining and running bioinformatics analyses. It provides data management to allow the user to reproduce analyses and to combine tools using a workflow system. BioWep (Romano et al., 2007) is another web-based application but only allows users to run pre-defined workflows to process bioinformatics data as there is no facility for creating workflows.

Publishers such as Springer now "*additionally encourage or require authors, as a condition of publication, to include in some article types a section that provides a permanent link to the data supporting the results reported in the article*" (Springer, 2014). The NIF (Marenco et al., 2008) and the Neuromorpho.org (Ascoli, Donohue & Halavi, 2007) portals both allow users to link publications with data described in those publications. More multifaceted publication repositories are under development that allow data and software to not only be stored, shared and associated with academic publications, but the software can also executed in situ. Execution of software and workflows takes place on cloud computing architectures. Elsevier Publishing are currently beta testing the Collage authoring environment (Collage authoring environment, 2013; Nowakowski, et al., 2011) which implements a cloud-based framework for executable papers. The papers must be written in a special mark-up language but authors can attach executable code and data to them. The code can be run by other users of the system in a web browser using the attached data or their own data. The system allows command line code to be run via scripts written in Ruby, Python, Perl or the UNIX shell (bash). The code is divided into modules (snippets) and these modules can be chained into pipelines using the scripts. RunMyCode (RunMyCode, 2013) allows users to create websites that complement academic papers where they upload their own data and software written in Matlab, R or WinRats. Other users can then run the software via a web browser. Users can supply their own data and parameter values.

The CARMEN VL interactive publication facility is aimed at the neuroscience community. Papers do not need to be written in a bespoke markup language which is specific to one platform. Thus, neuroscientists can continue writing publications in their desired format (such as PDF) using their preferred document preparation software (such as MS Word or LaTeX) and hosting them on their preferred platform where the paper can be linked from the VL. A much broader range of programming languages are available compared to the platforms just described and it allows users to compose scientific workflows. Furthermore, the individual

services in the workflow can be written in different programming languages and run on different operating systems as described later.

3 The CARMEN VL Infrastructure.

We firstly provide a brief overview of the existing CARMEN VL before describing the new publications extension. The VL is a distributed online platform for the secure storage and sharing of generic experimental or simulation data, metadata (structured descriptions of data, (Jessop, Weeks & Austin, 2010)) and source code. The VL is accessed via a web browser. It is publicly accessible, new users simply need to register at **https://portal.carmen.org.uk** (see section 9 for more details). The architecture is a 3-tier web architecture (Eckerson, 1995) and an overview is shown in Figure 1.

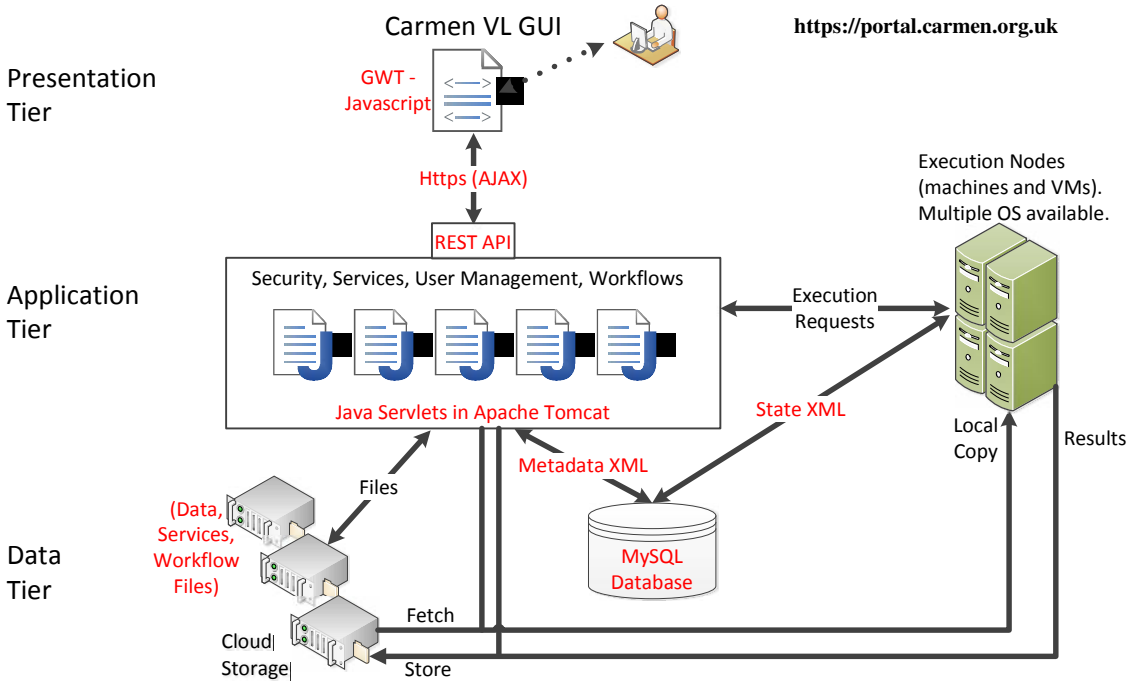

Figure 1. The underlying CARMEN VL Architecture which is a standard three tier web architecture (Eckerson, 1995). The Presentation tier GUI interfaces to the Application tier Java servlets over https via a REST interface. The servlets update the GUI using AJAX (asynchronous) updating. The Application tier servlets organise the processing of data, services, workflows and publications and orchestrate all communication within the VL. All data, software and workflow files are stored in the Data tier on cloud storage. The MySQL database in the Data tier stores the state of the system and its resources. The database also stores a constant log of all VL activity. The software services and workflow services run on the execution nodes or virtual machines on these nodes where a range of different operating systems are available to meet user's requirements. Data and software are copied from the cloud storage to the VMs and nodes to create local copies for fast access. All results from software execution are stored back in the cloud storage.

Presentation Tier: The VL provides a browser-based GUI so that users do not need to install software on their local machine. During the requirements gathering phase for the VL, users stated a clear preference for not having to install any software. To ensure cross-browser compatibility, the front-end GUI has been developed and implemented using the open source Google Web Toolkit (GWT) (Google web Toolkit, 2013) as an AJAX (Asynchronous JavaScript and XML) web application. GWT permits the application to be developed and tested in Java using standard software engineering practices and then compiled into cross browser JavaScript for use in any browser. AJAX enables asynchronous communication

between the Presentation and the Application Tiers so the VL can update parts of the GUI without reloading the whole page. We have ensured that the VL is compatible with a broad range of browsers, operating systems and devices (including tablet computers) and provides a very similar user experience on each.

Application Tier: A set of Java servlets running under Apache Tomcat (currently v5.5) which interface between the Presentation tier and the Data tier. The Application tier uses function calls over https for secure communication with the Presentation tier through a Representational State Transfer (REST) type interface using the four HTTP methods GET, POST, PUT and DELETE (Fielding, 2000). REST scales very well as the client and server are very loosely coupled and they can communicate over https so it is ideal for a distributed processing platform. The Java servlets orchestrate the system functionality (for example, data upload to allow data to be uploaded to the VL so it can be attached to a publication, running services that are attached to a publication or searching for publications) and communication between the various VL subsystems. When linking publications, the author inputs the details of the publication into a form and attaches data, service and workflow resources using the GUI. This information is converted to XML in the Presentation Layer and transmitted over https POST to the Java servlets which store it in the Data tier for future reference. Users can upload the software used in publications and the Java servlets allow users to run these Software as a Service (SaaS) in a cloud-based architecture (Armbrust et al., 2010). The cloud architecture provide scalable and dynamic deployment, provides high availability, enables simple system management and allows multiple operating systems (OSs) to be available as virtual machines (VMs). Services can also be linked into workflows using a visual workflow environment in the Presentation tier GUI. Again, the workflow execution is orchestrated by the Java servlets. The workflow execution servlets even allow services written for different OSs to be linked (this facility is detailed in section 4.4). Users can share their data, services and workflows with their choice of users and groups of users. A *security servlet* applies security policy, authorisation, and authentication.

Data Tier: The data tier contains a MySQL (currently v5.0) (MySQL, 2013) database to index *metadata* that describe *data, publications, services* and *workflows* together with state for service and workflow execution so all metadata and system logs can be preserved. The Data Tier also uses federated compute and data resources at the University of York, which form part of the White Rose Grid computing infrastructure (WhiteRoseGrid, 2013). Users' data and metadata including the data and metadata associated with publications are stored in a logical distributed file system which has a single global logical namespace and file hierarchy for ease of access.

The current hardware platform comprises a series of blade compute elements: 3 physical compute servers (8 cores, 16GB RAM) and 5 physical compute servers (24 cores, 112GB RAM) with a number of VMs (each currently 8 cores, 16GB RAM, a mix of Windows and different Linux distributions). There is 60TB of storage including 45TB for user data. We recommend that users do not exceed 500 GB of uploaded data and 100GB of derived data (from analyses) to encourage fair usage of the system for all users. The VL employs extensive virtualisation of its resources . It can be expanded by installing new storage disks for the cloud storage user data store, new execution machines with new VMs or new disks in the existing execution machines coupled with new VMs on these disks

3.1 CARMEN VL Publications Objectives

It is vital that a user-oriented system such as the CARMEN VL repository for interactive publications meets the expectations of users. Nowakowski et al. (2011) state a number of objectives for interactive publications. We use similar objectives and state how we meet those objectives in Table 1.

Table 1 Table listing a set of objectives for interactive publications and how the CARMEN VL publications meets those objectives.

Objective	How the CARMEN VL interactive publications meets the objective
Executable	The VL allows owners to upload services and associate these with a particular publication where they can be executed by any other user. Running a service allows the user to select input parameters, input (data) files and output files so a number of executions can be run and stored for comparisons. All results are available to the user in their file space. The owner of the service can also create example runs where the parameters and files are pre-populated to demonstrate the service to other users.
Reproducible, Provenance	When any service or workflow linked to a publication is run, the VL creates an XML log of all aspects of the execution. This allows services and workflows to be reproduced and repeated. The VL generates audit trails using extensive metadata. It currently records all actions taken within the system. These data can be mined and the results presented to a user providing information regarding: how the data, service or workflow has been used within the system; who viewed it, who ran a service or workflow; when they ran it and what the results were.
Standards	All metadata stored in the VL uses XML. This allows cross- compatibility with International Standards where applicable. Standards are under development for data archival (DataCite Metadata Working Group, 2011; Yarmey & Baker, 2013) and we already use the MINI standard (Gibson et al., 2008) for data storage. Once any standards have been finalised, then the VL can easily adopt these new standards as the XML is inherently flexible.
Software Compatibility	Services linked to publications run on the cloud infrastructure that provides a number of OSs available for execution as required by the service. Services can be written as Unix or Windows shell scripts, binary executables, or interpreted code (Matlab (as a compiled executable), Octave, Perl, Python, R, Ruby or Java). Data are stored in a distributed file system and retrieved by the VL as required.
Reference Compatibility	The publication details are stored in the VL in XML using a BibTeX-style schema. This makes the VL compatible with reference formats such as BibTex, EndNote and Refworks. Data can be imported and exported in these formats using a simple Java conversion to and from the XML.
Validation	When a user uploads a service they can associate test data with the service. This test data can be used to validate the service in the VL. We recommend that all users who are creating publications validate their data, services and workflows prior to constructing the publication to ensure correctness and reproducibility for other users accessing the data, service or workflow.
Licensing and Copyright	The VL is able to store and execute services written as binary executables, shell scripts or interpretable code. Users are bound by any licensing restrictions applicable for software and tools. Only software and tools that explicitly permit use on the VL can be used. Users are also bound by any copyright or licensing terms applicable to data. Many publications use data from common repositories. While some repositories allow the data to be hosted anywhere, other repositories place strict licensing conditions. The CARMEN VL requires all users to abide by these copyright restrictions.
Computation	The VL provides federated access to large compute clusters to run services and workflows linked to publications. The VL is hosted on a multi-cluster system at the University of York. . The system is designed to map jobs onto geographically remote machines which could be the White Rose Grid (a multi-cluster system shared between the Universities of York, Leeds and Sheffield) or other major supercomputer resources.
File Size	Because the VL hosts both a user's service and the associated data, it allows other users to run services using large data files without the expense of downloading the files for local analysis. All analyses can be started via the user's web browser and the service will execute on the VL execution nodes and virtual machines.
Access	Users can access any data, service, workflow or publication where the owner has granted

	them access rights. All data, services and workflows have permanent URLs (Life Science Identifiers (LSIDs) (Clarke, 2003)) and can be bookmarked within the VL. Users can elect to only share the metadata about the data or service but not the actual data or service. This effectively informs other users that the software or data file are present. Other users can then request access from the owner. The VL enforces access rights at the *user*, *list* of users, *group* or *all* level
Collaborative development support	Users can grant access rights to other users when they upload data, upload services, create workflows or create publications. When creating a workflow or publication, any data, services or workflows belonging to any user can be incorporated as long as the creator of the workflow or publication has sufficient access rights.
Multi-User environment	The VL is a multi-user archiving and execution virtual environment for neuroscience experiments and publications. The VL uses a distributed hardware platform allowing simultaneous service executions by multiple users.
Multi-Functional approach	The VL provides a scientific workflow tool to provide a systematic and automated facility for executing analyses linked to publications across different datasets; an easy and reproducible way of capturing the processing flow; and, the flow can be replayed, shared and adapted. The workflow allows services written for and running under different OSs to be combined.
Security	Users login to the VL through a secure login facility guarding access to sensitive data or computations. All data, services, workflows and publications are restricted to access by those users granted access rights by the resource owner.
Customisable	The VL is one of a number of related projects. The VL system allows the creation of separate and private projects which usually have their own URL and colour scheme and layout (skin). Such projects exist on the infrastructure but are self-contained and appear completely separate and are not visible to users outside of the project. Additionally, each user has their own view of the VL when they login which they can customise. The VL metadata schema is editable, allowing users to create templates particular to their datasets.

4 Author - Creating an Interactive Publication

In this paper, we use the term "author" to refer to the VL user who is creating the publication entry. This can be an author of the publication or another user registered with the VL with access privileges to the required resources. The CARMEN VL allows authors to link publications with data, services and workflows. Other users of the system can then view and execute the resources associated with a publication allowing reproducibility and validation.

Creating a publication and associating resources is a simple multi-stage process. The VL interactive publication feature does not require the paper to be written in a specialist and bespoke markup language. This means that the VL publications are backwards compatible and authors can add any of their existing or new publications and associate them with their data, software or workflows. The VL provides a hyperlink to electronic copies of papers. These can be hosted anywhere including on the author's website, other neuroscience repositories such as those described in section 2 or publishers' websites. The VL does not currently host the PDF documents as many journals and some conferences place strict copyright restrictions on hosting PDFs on websites. As journals move to open access allowing authors to host their PDFs more widely, then CARMEN VL will allow users to upload the PDF.

The interactive publications facility is a new feature in the CARMEN VL. New publications are being added by users as they become available. We use three papers that have been added to the system as exemplars in this paper (see figures 3, 5 and 7 in particular).

In the following subsections: we describe the metadata we use to describe the components of interactive publications and control the system, communication and processing in section 4.1, how to upload data in section 4.2, how to upload services in 4.3, how to generate a workflow in 4.4, section 4.5 describes how the user can set access rights for the publication and its associated resources, and finally, 4.6 details how to assemble the units and create a publication entry with associated data, software, services and workflows. We recommend uploading all data, software as services and creating all workflows prior to generating a publication entry in the VL. This allows the software services and workflows to be checked by the owner for runtime errors in their software and validated by the VL system. Any errors can then be corrected before generating a publication entry and the owner can be confident that their software or workflow will work as expected when other users run it.

4.1 Metadata

The CARMEN VL uses metadata to describe data, services, workflows and publications. The metadata are also used to control execution and communication within the VL. Each metadata XML schema is embedded in Java servlet (Java class). By encapsulating the schemas in Java, we are able to vary the validation levels. All user entered data are checked thoroughly but data generated automatically by the system are not checked.

We describe the four metadata formats next as they interlink. When publication entries are created and data, services and workflows are associated with them by authors, then the VL links the publication metadata with the metadata for the associated data, services and workflows.

Neuroscientists generate and process large data files during their experiments. Metadata are often scattered (Teeters et al., 2008). Data analysis and reproducibility require all of this information in a single, structured and organised file. Lawrence et al., (2011) state that metadata are vital to make data useful and Gray et al., (2002) assert that "*data is incomprehensible and hence useless unless there is a detailed and clear description of how and when it was gathered, and how the derived data was produced*". Hence, when users upload data to the VL, they complete a form which elicits information about their data and the experiments used to generate the data. The contents of this form are converted to XML metadata. For data file metadata, the VL uses the Minimum Information about a Neuroscience Investigation (MINI) standard (Gibson et al., 2008) for neuroscience electrophysiology experimental data. This metadata includes: name, description, ID, data type, and details of the neuroscience experiment used to generate the data. To provide meaning to the data and to allow other neuroscientists to interpret the data then the exact acquisition parameters and experimental conditions must be tracked and recorded as an audit trail of metadata (Ascoli, 2013). The experiment details logged include: the experimental context, the subject being studied, the anatomical recording location, any tasks performed, any stimulus provided, any behavioural events observed, recording protocol and data (time series) obtained. For output results files, the metadata includes: the name of the service used to produce the results, the service ID, the service type, the service parameter settings used, the OS platform, when it ran and how long it took among others.

A back-end Java servlet (Java class) validates the contents of the XML using a Java encoding of MINI. To make metadata entry simpler and quicker, any user uploading data can create pre-populated templates so that only the fields that change on a regular basis for a particular experimental domain need to be completed. As the VL uses XML to describe data, the XML format encoded in a Java class can be extended to any new industry standards for data

description (Yarmey & Baker, 2013), such as the proposed XML schema (DataCite Metadata Working Group, 2011). Adding a new metadata schema requires adding a new Java class to encapsulate it so, in this way, new schemas can be added and existing schemas remain in place.

The XML metadata describing services encapsulates all details of the service and its execution. These metadata are used by the system to both create the VL's user interface for services and to configure the deployment of the service onto the appropriate CARMEN execution node. The metadata is automatically generated by the system on service creation. The services metadata format was designed to hide the underlying VL technology for service execution from the user; provide keywords to the VL services search facility to enable discovery of relevant services; provide information to the user on the service, its usage, and its input and output parameters; allow the front-end to display the service and its execution requirements to the user; allow services to be dynamically targeted towards specific platforms (OSs and environments); and, configure the service parameters, input and output files and any software applications needed when running a service. The service metadata contain: the service name, a list of descriptive keyword supplied by the user, the input and output data formats; the algorithm used; details of any parameters supplied at runtime; and, the system components needed to run the service such as OS, OS version, processor, software interpreter, etc.

Workflow description metadata is produced automatically by the system when the workflow is created to store all details of the workflow and how it is to be executed. The workflow metadata description follows a similar form to myGrid's SCUFL (Simple Conceptual Unified Flow Language) script (Oinn et al., 2004), but modified to suit the CARMEN VL data and services. The processing in the workflow is therefore encapsulated by metadata that are high level, conceptual XML with each service (workflow step) represented by one atomic XML unit, inside one XML tag pair <service></service>. This simplifies parallel execution as the XML for each step is self-contained inside the tag pair so can be executed independently on separate execution nodes as all of the information required for execution is contained inside the tag pair. The metadata XML maps easily to Java classes so integrates easily with VL back-end workflow servlets which are written in Java.

To allow reproducibility, the system also transparently captures all the steps in the workflow execution process: the provenance for both data and services (Davidson and Freire, 2008; Freire et al., 2008). The VL keeps a complete audit trail using the Workflow metadata XML that contain full information regarding all of the service version numbers, where each service ran, what the inputs/outputs were, start time, execution time, whether the workflow services succeeded and the workflow version. Thus, the VL can re-run a workflow instance from the metadata allowing full reproducibility. Davidson and Freire (2008) state that workflow provenance "*provides important documentation that is key to preserving the data, to determining the data's quality and authorship, and to reproduce as well as validate the results. These are all important requirements of the scientific process*."

Publication metadata in XML is generated when the user has entered the publication details and associated all necessary resources. Again, the metadata are captured by a Java class in the back-end servlets which validates the text extracted from the user entered fields. The XML describes the paper details and lists the associated data, services and workflows. The metadata format is based upon common reference formats, it is mainly derived from BibTeX (Mittelbach et al., 2010) but we ensured that it met the EndNote, RIS, RefMan and RefWorks

formats too for cross-compatibility. This means that any publication data in any of these formats can be mapped into the VL. This will simplify the development of future functionality such as reference import and export proposed in the future work section of this paper by allowing easy mapping to common reference database formats. The metadata feeds into the VL publications search function so there are a number of required fields used in the search indexing that are common across all publications (author, title, year, keywords, abstract, DOI, URL and the linked data, services and workflows). The remaining sets of fields vary per publication type as in BibTeX.

4.2 Uploading data

Many neuroscience publications describe data generated by experiments, generated by software or used as input to neuroscience software algorithms. Therefore, it is vital that authors can link this data to a publication so other users can explore the data interactively. Registered users can upload experimental data to the VL, associate extensive metadata and link this data and metadata to publications. The CARMEN VL currently hosts 547,546 neuroscience data files (11 Dec 2013). These data files include electroencephalography (EEG) data; Magnetoencephalography (MEG) data; Multi-Electrode Array Analysis (MEA) data of retinas; physiological tremor data recorded by local field potential (LFP) of finger movements and electromyography (EMG) recordings of forearm and hand muscles; voltage sensitive dye image series using high power laser illumination of auditory cortexes; and, high luminescence laser imaging of rhythmic activity in neurons.

4.2.1 File Upload

Data are uploaded using a Java applet running within the browser. The applet applies error checking and can handle large files by invoking multiple, parallel streams over multiple HTTPS ports. Once a data file has been stored, it can be searched; the metadata viewed and edited; annotations added and viewed; downloaded; visualized; and, analysed by user supplied code.

4.3 Uploading Services

Many neuroscience publications use existing algorithms to process data, describe new algorithms or extensions to existing algorithms. However, a simple publication repository does not allow access to this software. CARMEN VL users can upload their own software to create an executable Software as a Service (SaaS) (Weeks et al., 2013). This allows an author to link software to a publication entry so other users can investigate and run the software against the data used in the publication or against any other data stored on the VL. For the author, this is all achieved via the VL's browser interface, and all data and processing are contained within the VL platform. We surveyed a set of neuroscientists and ensured that the VL is able to handle the full range of programming languages and shell scripting languages commonly used by those neuroscientists. Software written as **Unix or Windows shell scripts**; any programming language that generates a standalone executable such as **C, C++ or FORTRAN**; or, any of the following interpreted languages: **Matlab** (as a compiled executable), **Octave, Perl, Python, R, Ruby or Java** can be uploaded and run as service. Additionally, the survey identified a need for an executable software platform that requires minimal knowledge of the underlying service execution technology.

4.3.1 Service Builder Tool

A VL service consists of a (slightly modified) command-line application that is wrapped inside a Java dynamically-loadable class, and embedded into a JAR file. To allow the software detailed in a publication to be deployed, the VL provides a Service Builder tool; a Java application that hides the wrapping process (Weeks et al., 2013). This is a standalone

desktop application which is used to bundle algorithms, source code and test data together with metadata into a JAR (Java ARchive) file and Java wrapping code. The Service Builder uses text-based forms to gather information from the user who is creating the service, such as input files, parameters and output files, and this information is stored in the service's metadata XML. The VL automatically uploads the service bundle to the system where it is stored and registered as a service using the metadata. The developer sets all access rights for their service. Once registered, the service is available for execution and can be linked to a publication as described in section 4.5.1.

Once a service has been uploaded, the owner of a service can create example runs of their service in the VL that demonstrate the operation and capabilities of the service to other users. These examples have all the input parameter values and file names pre-set, typically to demonstrate some interesting results, or to reproduce analyses from a publication. Multiple examples can be created for each service.

There is currently a rich array of neuroscience services available on the CARMEN VL to be run as individual software-as-a-service processes or combined into workflows. Many of these services have been written by neuroscientists. A number of filters are available to prepare the data (low pass, high pass, Butterworth (including filters that can handle multi-channel time-series data), finite-duration impulse-response equiripple filter). Spike detectors range from well-known techniques (e.g. simple thresholding, wavelet based (Wave_clus) (Quiroga et al., 2004), energy-based such as the nonlinear energy operator (NEO) (Mukhopadhyay & Ray, 1998) to techniques based on statistics (Masud & Borisyuk, 2011) and higher order statistics (Shahid, Walker & Smith, 2010). Once spikes have been detected they can be sorted using unsupervised clustering of multidimensional continuous data into a mixture of Gaussians (the algorithm is known as Klustakwik) (Harris et al., 2000). Other services allow users to analyse spontaneous activity patterns recorded from retina cells by Multi-Electrode Array Analysis (MEA) (Eglen et al., 2014; Sernagor, 2011; Simonotto et al., 2009; Simonotto et al., 2011). The Multiple Interacting Instantiations of Neuronal Dynamics (MIIND) (de Kamps and Baier, 2007) neural simulator has been wrapped and installed as a service. A Granger Causality service has been uploaded which quantifies the influence of one time-series on another time-series where the time-series can be multidimensional (Zou et al, 2010).

4.4 Creating a Workflow

Analysing research data and the evaluations described in publications frequently requires a number of execution stages to be performed in series. Common steps include data pre-processing, feature selection, model generation and model testing and validation. Many of these individual steps are iterative. The CARMEN VL provides a scientific workflow environment to combine software services into a processing tree (Ingram et al., 2012). The workflow even enables services written for and running under different operating systems to be combined. Workflows can be linked to a VL publication where applicable and other users can run these workflows to recreate the experimental analyses described in the paper. The workflow provides systematic and automated analyses across different datasets; an easy and reproducible way of capturing the processing flow; and, this processing flow can be replayed, shared and adapted.

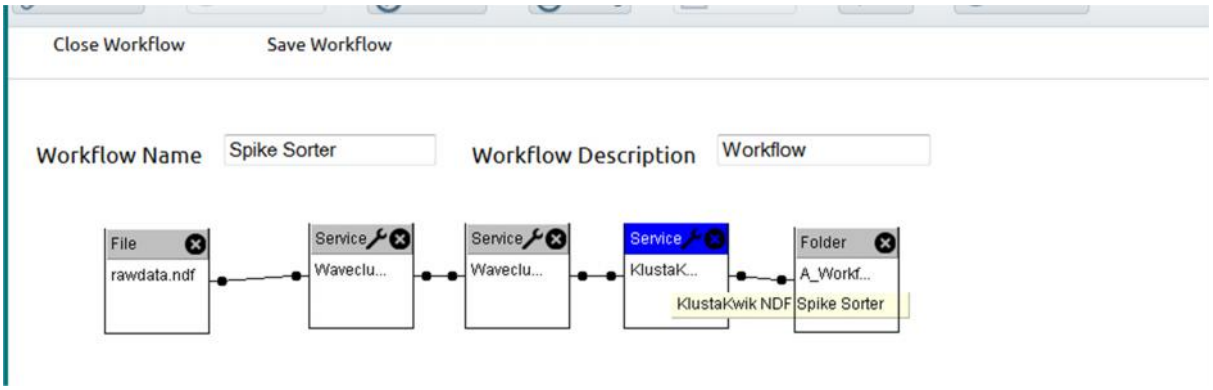

Figure 2 Screenshot of an example workflow for spike detection. The data feeds into a Wave_clus high pass filter, followed by a Wave_clus spike detector (Quiroga et al., 2004), the spikes are sorted using a Klustakwik spike sorter (Harris et al., 2000) and the results are output to a selected folder.

An overriding issue with workflows is enabling data transfer from one step of the workflow to the next. Data formats must be compatible along the workflow to allow input and output of data. Neuroscience data are stored in a variety of data formats (Herz, et al, 2008) which makes sharing and analysing data difficult. The CARMEN project has developed a standard data format that allows heterogeneous data to be specified and encapsulated in an XML wrapper: Neurophysiology Data translation Format (NDF) (NDF, 2013) which is built into the CARMEN VL. An NDF dataset consists of a metadata/configuration file in XML format and references to the set of associated data files (Liang et al., 2010). The metadata uses the MINI standard (Gibson et al., 2008) which is also used in the CARMEN VL to describe data (see section 4.1). Hence, metadata can be transferred easily from VL to NDF and NDF to VL. NDF can encapsulate multiple data formats within a single container including image/video data, time-series signal data and segmented time-series data. This allows data to be processed in a consistent way and provides a standard for sharing data, specifically for data transfer between the VL services. NDF allows selection of specific channels and time windows within multichannel data sources. It is particularly useful for workflows to ensure that the output data of one service will be compatible with the input data requirements of another service. The VL has conversion services to allow users to convert to and from the NDF format as required. For example, a conversion service which takes Multi-Electrode Array (MEA) data in HDF5 format and converts it to NDF is available and forms a service in the workflow shown in Figure 3 and (Eglen et al., 2014).

For workflows, we evaluated the Taverna (Hull et al., 2006) and E-Science Central (Woodman et al., 2009) systems. However two specific areas created insurmountable problems; presenting an integrated environment with a common interface, and integrating CARMEN's NDF data format. Hence, we developed a bespoke workflow environment tailored to the CARMEN VL but using these other workflow tools as guides. The VL workflow tool is Java-based graphical workflow design environment running in the VL window and a back-end workflow execution engine with access to a library of services and common workflow tasks. The graphical design tool uses simple drag, drop and connect operations. It is easy to use requiring no computer programming knowledge. Each component in the visual interface represents a software service. These services can be linked using drag and drop. When linking services, the GUI ensures that the output of a previous service is compatible with the input of a successor service before they can be linked. This enforces data compatibility and will ensure correct execution of the workflow. Thus, complex computational processes can be created as in Figure 2 and Figure 3.

Figure 2 shows a neuroscience workflow comprising: one input data file of Multi-Electrode Array (MEA) data; three services in a pipeline to 1) filter the data, 2) detect spikes in the data and 3) sort (cluster) the detected spikes; and an output folder where the results of the processing will be output by the VL. In this example, the output would be a sorted list of spike indices from the original MEA data. The workflow in Figure 3 is taken from the publication (Eglen et al, 2014) and comprises four parallel pipelines. Each pipeline processes a different input file which contains MEA data from Advanced Pixel Sensor arrays. Each pipeline has two services. The first service converts the data files which are stored in Hierarchical Data Format 5 (HDF5) format (HDF Group, 2000) to the NDF format required for the input to the second service. This second service finds bursts of activity from spike times, for each channel in a MEA (Eglen et al., 2014). When this workflow executes, each of the four pipelines will execute in parallel on the execution nodes.

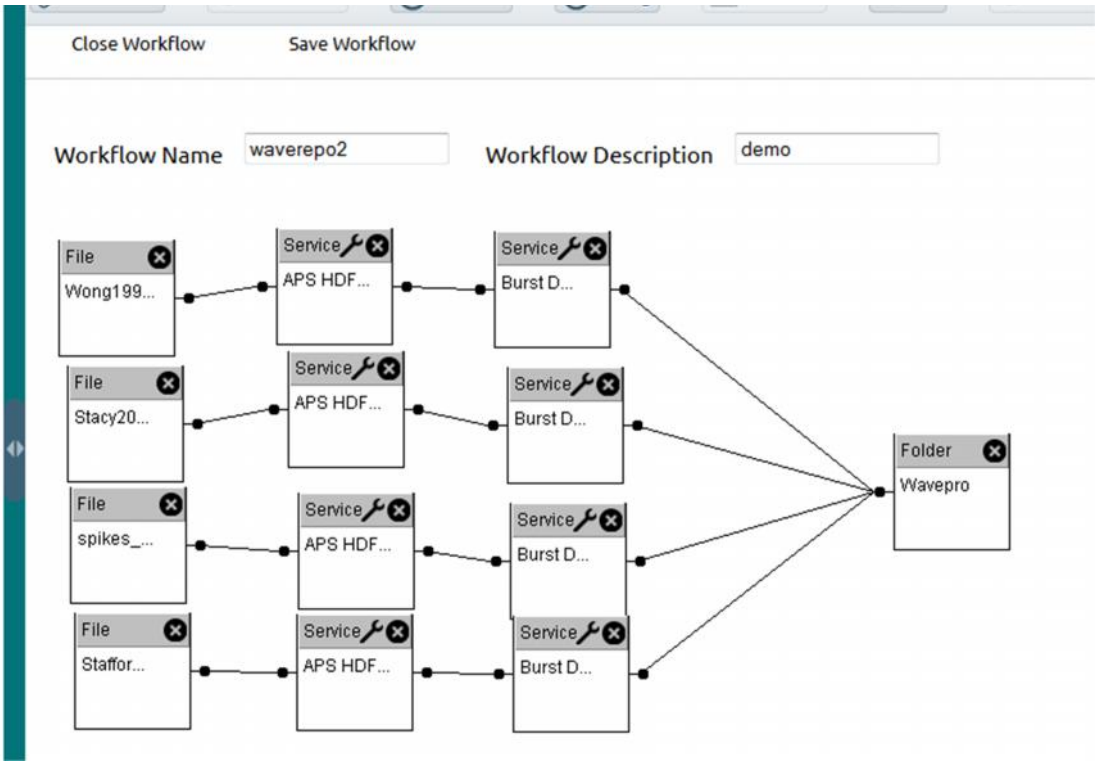

Figure 3 Screenshot of a workflow used in a publication (Eglen et al, 2014). Four different data files form the input to a pipeline of two services- a data converter and a burst detector. The VL workflow engine will execute the four separate pipelines in parallel with all output files in a single folder.

The workflow execution is underpinned by an assortment of service-execution infrastructures so this allows each workflow to transparently mix services from differing operating systems and application programming technologies. An example of how the workflow in Figure 2 might be implemented using different OSs and programming languages is shown in Figure 4. The high pass filter could be written and compiled for Ubuntu Linux using C++. The spike detector could be a MATLAB for MS Windows executable and the spike sorter could be written and compiled using C under CentOS Linux. The VL uses virtual machines on the execution nodes running the appropriate OS to execute each step of the workflow and passes copies of the data to and from the VMs as appropriate. Full details of how the VL executes a workflow across multiple platforms are given in section 6.3.

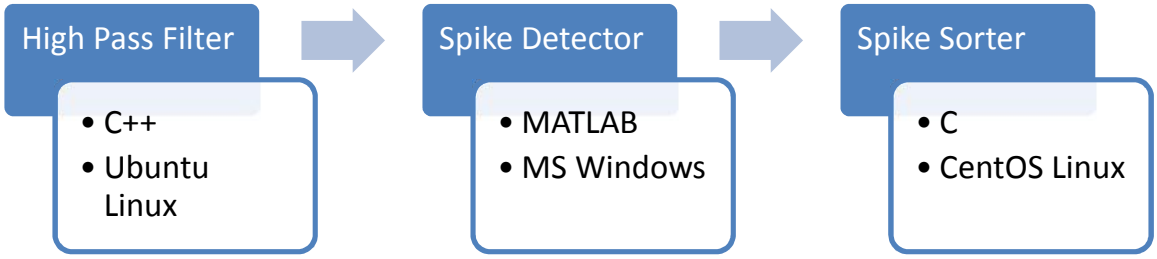

Figure 4 Showing the three services from the workflow in Figure 2 with an example of how these services could be written in different programming languages and run on different OSs.

As users add services to the VL, then a set of common processes will be available for inclusion in workflows (see section 4.3.1. for examples). These can be incorporated by any user into their workflow. Littauer et al. (Littauer, et al., 2012) posit that it is important for workflow environments to provide libraries of components which perform simple tasks. These components can be easily added to user's end-to-end workflows. Once a system has established a library of workflow and workflow components then new users will be encouraged to build workflows using the library of components and using other user's end-to-end workflows as templates.

4.5 Sharing

CARMEN encourages users to share data and services with all other users to promote collaboration. However, this is not always possible due to copyright, licensing or privacy restrictions which must be adhered to. Therefore, when a user uploads data, a service or creates a workflow, they are asked to choose the sharing settings for that resource and its associated metadata. These sharing settings can be set to; ***all*** such that anyone with an account for the VL can view it which is recommended unless copyright or privacy restrictions prevent this; ***user*** (private) such that only the owner of the resource can view it; ***list*** such that only a subset of users can see it - these users are selected by the resource owner from a dropdown list of all users; or ***group.*** The CARMEN VL allows users to create collaborative groups and resources can be shared with all group members rather than just an individual. This precludes having to select individual users from a dropdown list for sharing but restricts access where restrictions apply.

4.6 Generating the Interactive Publications

A number of publication types are available in the CARMEN VL:

- journal article,
- conference proceedings,
- book chapter,
- chapter in a collection (with its own title),
- technical report
- thesis.

The first step is to enter the citation details in a form. Each publication is represented by a set of fields common to both the entry form in the VL and the metadata for the publication (see section 4.1 for metadata details). Some fields are compulsory as these feed into the VL "Publications" search mechanism to allow other users to be able to find and retrieve publications (described in section 5).

The CARMEN VL provides DOI (DOI, 2013) lookup using www.crossref.org to cross-reference publications. If the publication has been published and entered into DOI (DOI,

2013) then the user can simply enter the DOI name and the contents of the form will be automatically populated by the data retrieved from www.crossref.org. The user can amend and add to this data as required. Otherwise the data must be entered manually in the form. An example populated form is shown in Figure 5 for a journal article.

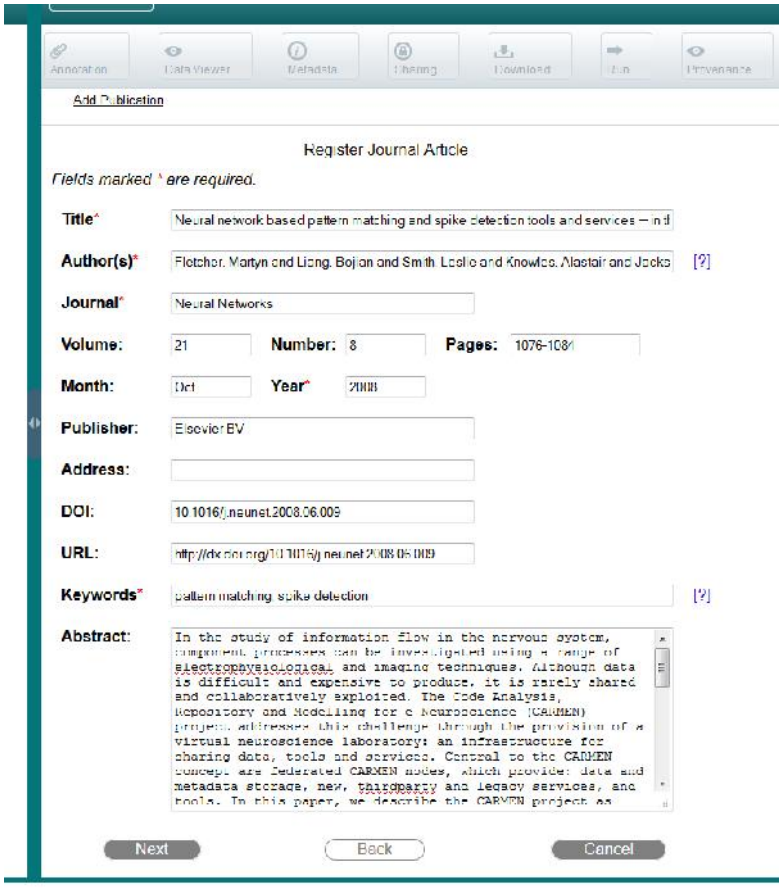

Figure 5 Screenshot of the form for a journal article populated by an article details. The author can enter a DOI into the form which is cross-referenced at www.crossref.org and the form is auto-populated using the results returned. Alternatively, the author can populate the form themselves.

Once the publication details have been entered, the user can associate data, services and workflows with a publication as detailed in the next section to make the publication interactive and reproducible.

4.6.1 Linking Data, Services and Workflows

In the CARMEN VL, data, services and workflows can have permanent links associated with them known as Life Science Identifiers (LSIDs) (Clarke, 2003). An example LSID is: https://portal.carmen.org.uk/#link=URN:LSID:portal.carmen.org.uk:metadata:204

This provides a permanent link to resources which can be easily cross-referenced in any publication, both publications within the VL or elsewhere. It also prevents duplication as multiple publications can be linked to each data file, each service or each workflow.

The process to link data is shown in Figure 6 and the process for linking services and workflows is identical. Both input data files and output (results) files from software services are stored on the VL allowing easy comparison and validation of the service by any user. Associating workflows to publications allows pipelining and parallel execution of services and these pipelines can also be investigated and executed by other users. Littauer et al.

(Littauer, et al., 2012) posit that the reuse of scientific workflows will be increased by raising awareness which can be achieved through citing them in publications, sharing them with other researchers and promoting them through electronic communication. Reuse and promotion will be facilitated if workflows have stable and persistent identifiers (the CARMEN VL LSIDs) which increase the lifetime of workflows long after they are published. Without these permanent links, workflows are temporary and likely to be lost.

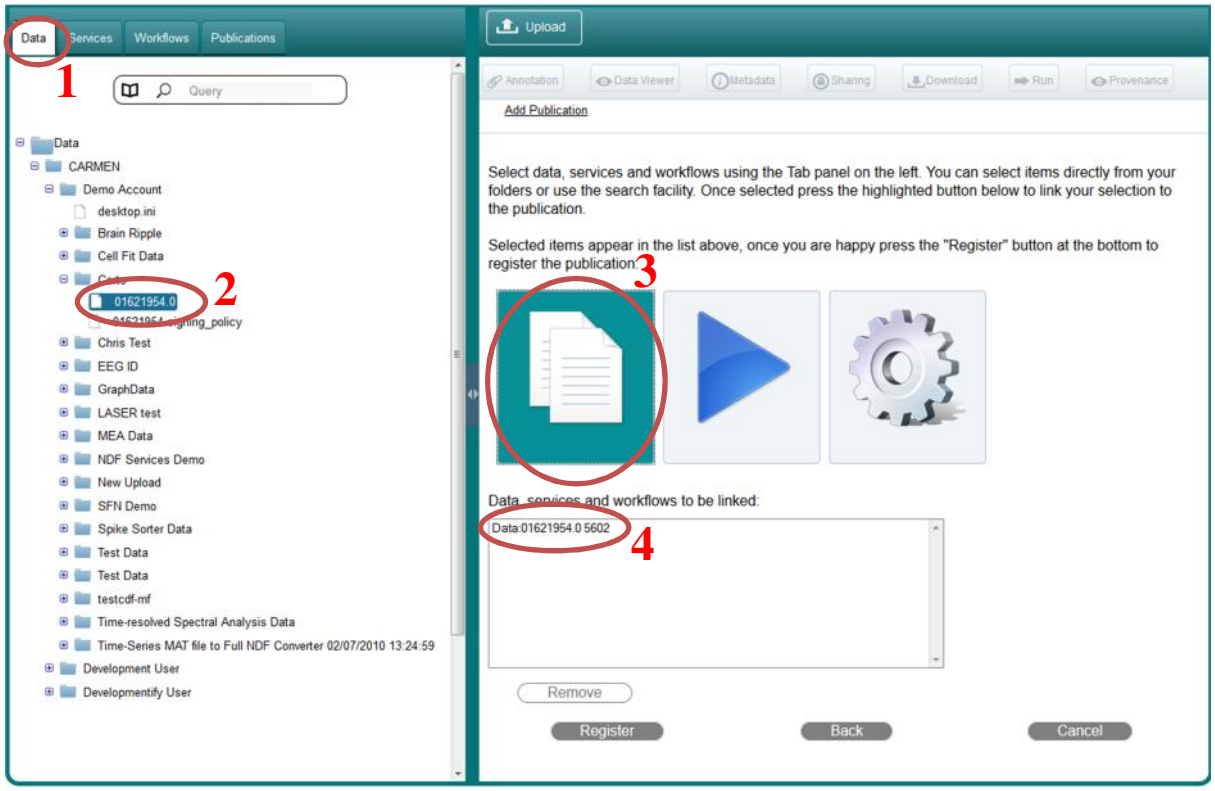

Figure 6 The CARMEN VL allows users to associate data, services and workflows with a publication. The figure shows a screenshot of the process for associating data. Users can associate data stored in the CARMEN VL by selecting "Data" (1), searching in data for their required file (2) and adding it to the publication (3). Users can check which data files are currently associated (4). Metadata describing data, services and workflows are also available to other users exploring the publication once it has been created to provide more meaningful information.

4.7 Storing Publications

On completion, CARMEN VL generates the publication metadata described in section 4.1 which is stored in a MySQL database. Thus, there is a complete record of each publication and its associated resources stored permanently in the database for future reference. The metadata are also input to an Apache Lucene-based search system described in section 5 to allow publications to be searched.

5 Users - Searching Publications

A search capability is vital, particularly as the number of data files, services, workflows and publications grows over time making any complete view of the resources unfeasible. Authors creating publications need to be able to search for data, services and workflows if they have not been bookmarked. Other uses can explore the publications using the search facility. The VL has four separate search facilities: data, services, workflow and publications and maintains a separate index for each.

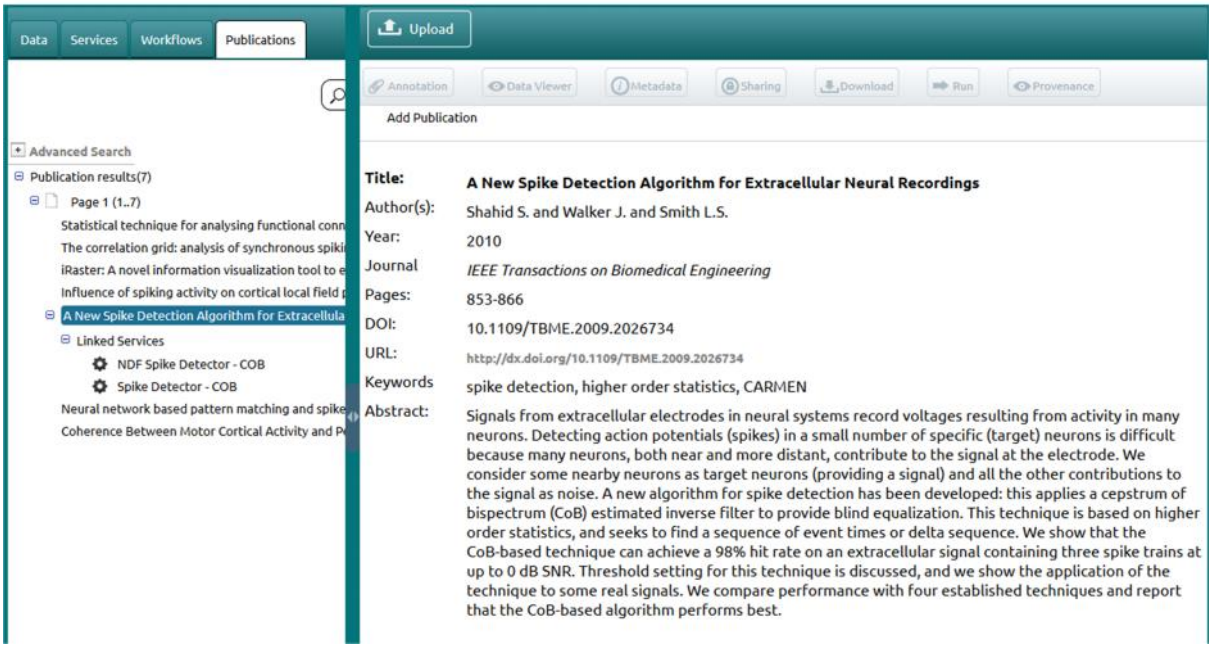

Figure 7 The CARMEN VL simple search finds any publications where the query terms match any field. The figure shows the results of a simple search for the term "spike". Seven publications match and are listed in the left hand panel. From these results, the user has selected the 5th match. Any associated data, services and workflows are displayed on the left and the publication details (authors, title etc.) are displayed on the right. This matching article has two software services associated: two spike detectors.

The CARMEN VL uses Apache Lucene (McCandless, Hatcher and Gospodnetic, 2010) to underpin the search: a feature-rich, mature and robust Java API with a large development community and wide industry adoption. The Lucene Java API is integrated with the Java search servlet in the back-end. The full Lucene query syntax is enabled providing: Boolean logic operators; term modifiers: wildcard searching, fuzzy searching and proximity searching; term boosting; term/phrase grouping; and, fielded searching. We have extended Lucene to include a "Did you mean?" functionality based on the "Did you mean?" functionality of Google search, autocomplete (suggest-as-you-type) functionality (Hodge et al., 2012) and synonyms. Autocomplete decreases the possibility of misspelled and redundant query terms by advising users of stored terms and also by minimising typing as the word or phrase is completed for the user. The VL synonym facility retrieves synonyms from the Neuroscience Information Framework (http://www.neuinfo.org/) dynamic inventory of Web-based neuroscience information (Gardner, et al, 2008). The user's query term is submitted to the NIF portal which returns a list of synonyms to augment the user's query. For example, if a user requests synonyms for *"glial cell"* in VL then NIF returns *"neuroglial cell" "glia"*.

The publications search has a simple search which matches publications against a set of query terms input by the user as shown in Figure 7. The search is performed on the Lucene index for publications data. There is also an advanced search feature which provides fielded search where publications that match the query terms in the specified fields will be retrieved. The fields available are shown in Figure 8: title, authors, keywords and publication title. These fields are the compulsory fields when a user enters the details for a publication (see section 4.1; and Figure 5) and are common across all publication types. They are stored as indexed fields in the Lucene search index allowing them to be searched individually.

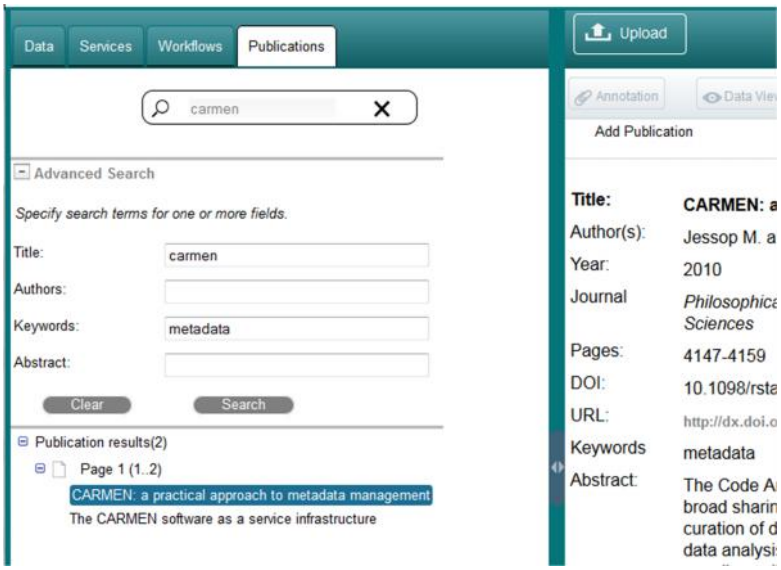

Figure 8 Advanced search allows the user to perform a fielded search. In the figure, the user has performed an advanced search for the term "carmen" in the title and "metadata" in the keywords. Two publications match.

6 Users - Exploring Publications

The main aspect of the CARMEN VL publications facility that sets it apart from other publication repositories is the interactive features. After searching for publications, clicking on a matching publication in the results opens that publication's details including author, title and publication year and lists the associated data, services and workflows (see Figure 7). From here users can explore publications, analyse the associated data and its metadata and reproduce and validate the publication's results through executing services and workflows.

6.1 Data Interrogation

Users can investigate the input data files and any output results files generated by services used in the publication to validate and reuse the results. Data files and metadata (see section 4.1) can be viewed by any user with access rights within the VL or files can be downloaded onto that user's computer for investigation. The details stored with both input and output data provide provenance and traceability for both authors and users. The user can also download the data file to their desktop and investigate the data from there.

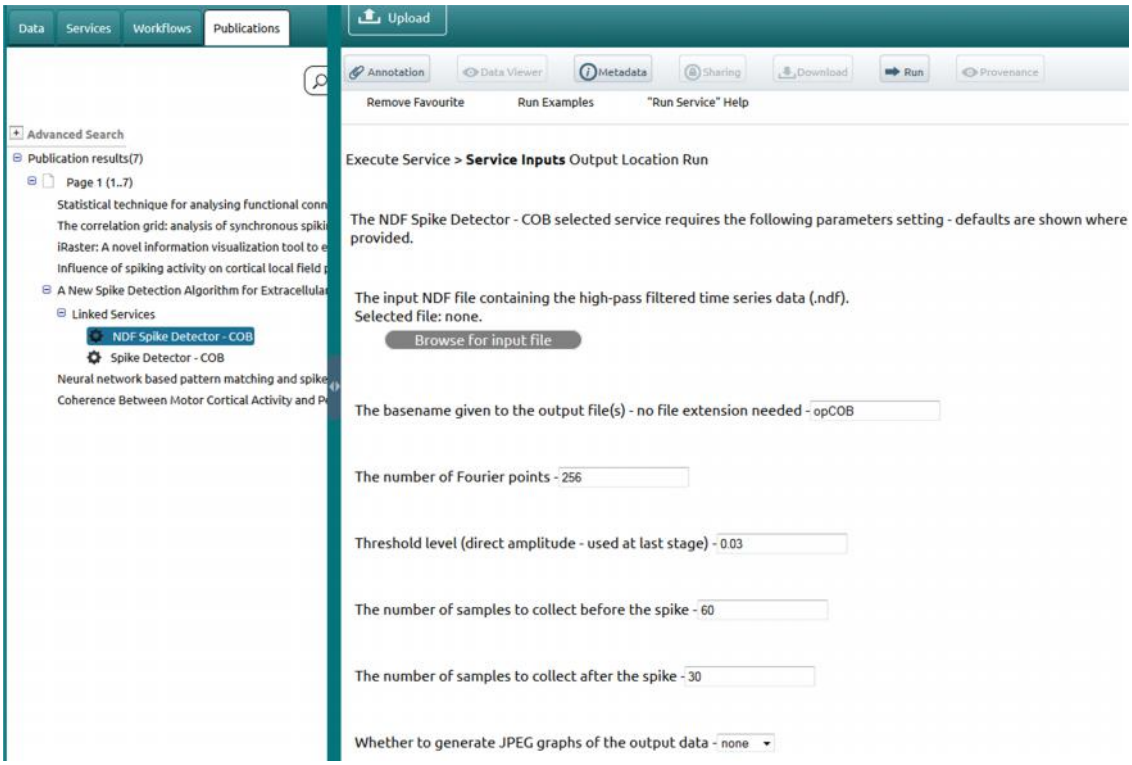

Figure 9 The CARMEN VL provides reproducible results. The service can be prepopulated with data files and settings. In this screenshot, the parameter values are set but no input file is specified. However, the service requires an NDF type input file so the user will only be able to select a file of this type to ensure correct operation of the service. When the user selects the service, they can then execute the service as it was executed for the publication.

6.2 Service Execution

The ability to execute services is a vital capability of the CARMEN VL. Additionally, other users can investigate the service further by uploading their own data onto the VL and running the service using this data. If the user decides to run a service linked to a publication then a panel is displayed, as shown in Figure 9. The panel contents are generated from the service metadata which contains all of the information about the service. Any service parameter settings, input file names or output file names can be prepopulated by the author during service creation. The fields in Figure 9 are pre-populated with filenames and parameter values. Alternatively, the user can choose to set the values. The metadata describing each service contains constraints for the service parameters. The metadata and thus, these constraints, are input to the service execution servlets in the back-end allowing the VL to ensure that any user selected parameter values are valid. The VL allows repeated execution of a particular service. Hence, users can choose to vary the parameter settings or the input data file to analyse the performance of the software over a series of runs.

The VL front-end GUI requests service execution using https requests to the Application Tier (see Figure 1). At execution time, the service manager servlet fetches the required input data files and the service's JAR file from the cloud storage. The Java servlet then passes processing to a job scheduler servlet which orchestrates the service execution on the system's distributed processing nodes using the service metadata information and the fetched input files. There is a pool of execution nodes for service execution which are able to use load balancing when executing jobs. The nodes can be executable compute nodes or VMs which can be created with applicable OSs to permit service execution and allow scalable and dynamic deployment in one or more clouds. The job scheduler selects the appropriate

execution environment at run-time based on the requirements of the service as specified in the metadata and copies the fetched service files and data files to the local execution node for fast local access during processing.

Once a service has been invoked, the execution servlets maintain a log which the VL GUI displays and regularly updates to show the user the current state of the service. These updates represent the execution steps and allow the user to monitor progress. The entire log is written to the user's file space when the service completes. The log provides provenance of service execution. Users can view the log and verify and analyse all stages of execution if they wish.

The service's metadata describes how to handle any output files produced by the service and these are stored back within the file system at the requested location. For example, a spike detection service may output a text file containing the indices of the spikes detected in a particular data channel and a graph file of the detected spikes in JPEG format (see Figure 9).

6.3 Workflow Execution

If the user requests a workflow execution then the workflow execution servlet creates a service invocation request, and passes it to the job scheduler servlet. Each workflow is described by metadata XML (see section 4.1). Using this metadata, the workflow environment enables parallel execution and manages both data and control flow between connected services. The job scheduler allows the workflow engine to make use of the VL's dynamic service deployment and execution system, achieving scalable heterogeneous distributed processing.

To enable cross-platform processing of mixed workflows, for instance the cross-platform workflow in Figure 4, workflows are submitted to a Workflow Engine (WE) servlet on one of the service execution nodes. This will be a node running an OS that is capable of running at least one, if not most, of the services in the workflow. In the example in Figure 4, this may be an Ubuntu Linux node. The WE executes the first service in the workflow and as many services as possible in parallel. For each subsequent service (after the first service and any parallel execution), the WE determines if this next service can be run locally on that node, or whether it needs to run elsewhere. If the service runs elsewhere, then the WE tests if the input data required by the service is local to the chosen node or in the system cloud storage. If it is local then the data file is uploaded to the cloud storage where it is available to any execution node and the service is submitted to the job scheduler servlet on the system. The job scheduler selects a suitable execution node with the correct OS for the service and which is not busy. When this service finishes, its output files will be stored in the cloud storage. As each service runs, the WE determines if it can run locally or needs to be submitted to the job scheduler. At the end of the workflow, the WE works out what data needs keeping, and deletes the rest. The data to be kept are stored in the user's area of the cloud storage repository. All local data is deleted to tidy the system by removing temporary files.

Previous workflow runs can be reproduced and investigated using the Workflow metadata to guide the processing. The workflow metadata captures all aspects of workflow execution (see section 4.1). Although the workflow layout and the flow of processing when the workflow runs is fixed, the settings available to the user when they run a workflow can be varied. The input and output files can be altered and parameter settings of the individual services can be varied.

7 Conclusion

The CARMEN VL is a web-based collaborative neuroscience virtual laboratory. It is underpinned by a cloud computing platform allowing multiple software services to be processed in parallel and large data sets to be processed and analysed. This paper describes the new publications facility for the VL. It allows neuroscientists to capture and share the results of neuroscience research projects and publications with particular emphasis on providing long-term access to the data, software and scientific workflows described in publications (Austin et al., 2011). In the VL, publications and their associated data software and scientific workflows are interactive and assured of long-term preservation and curation. Providing wider access to research results from electronic publications has both expedited and improved the effectiveness of follow-on research in many scholarly disciplines (Choudhury, 2008).

Users can create publication entries by: cross-referencing DOIs and importing the publication metadata; or, by entering the data themselves in a form. Data, services and workflows can be linked to the publication using internal URLs (the CARMEN VL uses LSIDs). The VL interactive publication feature does not require the paper to be written in a bespoke markup language. This means that the VL publications are backwards compatible, require no specialist language knowledge and authors can add any of their existing or new publications and associate them with their data, software or workflows. The VL provides a hyperlink to electronic copies of papers. These can be hosted anywhere including on the author's website, other neuroscience repositories such as those described in section 2 or publishers' websites.

Abiding by licensing agreements is vital. Users have full control over access permissions for their own metadata, data, services and workflows. The resource owner can decide whether to grant access rights at the *user*, *list* of users, *group* or *all* level.

The VL has been developed over a period of 6.5 years. The development process adds new features, many requested by users and from user requirements analyses and evaluations. The users analyse the system from aesthetic, technical, organisational and social perspectives. This feedback is used to refine and improve the functionality and refine the look-and-feel. The users' comments provide a roadmap for future developments for the interactive publications and the VL in general. For example, users can create publications at the moment but cannot edit them. There are various approaches for allowing editing: users can have a "homepage" that shows all of their publications and they can edit them from there or they can search for their publications using the VL search mechanism. We will elicit the preferred approach from the user feedback.

8 Future Work

Currently, a user can only upload source code as a zipped tarball of code for a software service that they are uploading and building on the VL. However, we plan to link source code with software services and publications in a more flexible way. We will investigate allowing users to associate code with software services and with publications and making source code viewable and potentially editable. The Ajax.org Code Editor (ACE) (http://ace.ajax.org/) an embeddable code editor written in JavaScript has recently been integrated with GWT (https://github.com/daveho/AceGWT) and would seem an ideal tool for this functionality. ACE features support for 45 languages, syntax highlighting, is able to display huge documents (100,000 lines and more) and provides regular expression matching, among others. Users would then be able to associate: data; services plus source code for those

services; and, workflows. The services popup could be enhanced with links to source code. Clicking on the link would open a new tab displaying the source code in the code editor.

Many publication repositories allow users to export citations usually in a number of formats. We propose adding a button to the publication display panel (see Figure 7). This would parse the publication's metadata which stores the publication's details using BibTeX style XML and display the citation in BibTeX, RIS format (for Reference Manager, ProCite, EndNote), RefWorks and text formats (Google Scholar (scholar.google.com) displays MLA, APA and Chicago). We will consult with neuroscientists to find the preferred formats. Users will then be able to cut and paste their selected format from those displayed.

As the use of the research identifiers ORCID (http://orcid.org/) and ResearcherID (ResearcherID, 2013) expand, we will investigate using them in the CARMEN VL to provide unique identification of users. "*Each member is assigned a unique identifier to enable researchers to manage their publication lists, … and avoid author misidentification*" (ResearcherID, 2013). This will also assist with the VL search mechanisms by providing a unique index key for authors so users can ensure they find the correct author when multiple authors have similar names.

Security issues in the VL are very important. It is easy to upload data, and to implement a service. This leads to the possibility of, for example, deploying malicious services. For software services, we are investigating testing frameworks using sandboxing that will overcome this. When services are first uploaded, they are placed in their own sandboxed virtual machine to reduce the possibility of cross infection. Once the service has been authorised (run successfully), it can be transferred to the VL system. Any service that is sandboxed and found to be malicious will be removed along with its VM.

Although the CARMEN Project is focused on neuroscience we are now porting the VL to new domains to work on generic data/software in the YouShare system. To allow this the metadata schema is editable, allowing users to create templates particular to their datasets.

9 Information Sharing Statement

The Carmen VL is a publicly accessible Virtual Laboratory (VL) for neuroscientists. New users simply go to the URL (https://portal.carmen.org.uk) and click the "Sign up and get started with CARMEN" button to register. All facilities described in this paper are available to all users. CARMEN encourages users to make their data, services and workflows publicly accessible but copyright and privacy restrictions may prevent this. Any restrictions must be adhered to when using the VL. Hence, owners can make data, services and workflows visible to *user*, *list* of users, *group* or *all*. Other users can access any data, service, workflow or publication where the owner has granted them access rights. All data, services and workflows are underpinned by permanent URLs (LSIDs) and can be bookmarked within the VL for future use.

10 Acknowledgement

The CARMEN VL was developed under funding provided by the UK Engineering and Physical Sciences Research Council (grant number EP/E002331/1) and development of the VL is now supported by the UK Biotechnology and Biological Sciences Research Council (grant number BB/I000984/1).

11 Dedication

We dedicate this paper to the memory of our dear friend and colleague Professor Colin Ingram who passed away on December 15th 2013. Colin was the lead investigator on the CARMEN project and co-author of this paper.

References

Aine CJ, Sanfratello L, Ranken D, et al. (2012). MEG-SIM: A Web Portal for Testing MEG Analysis Methods using Realistic Simulated and Empirical Data. *Neuroinformatics*, *10*, 141-158.

Armbrust, M., Fox, A., Griffith, R. et al. (2010). A view of cloud computing. *Communications of the ACM*, *53*(4), 50-58.

Ascoli, GA. (2013). Global neuroscience: distributing the management of brain knowledge worldwide. *Neuroinformatics*, 11(1), 1-3.

Ascoli, GA., Donohue, DE., & Halavi, M. (2007). NeuroMorpho. Org: a central resource for neuronal morphologies. *The Journal of Neuroscience*, *27*(35), 9247-9251.

Austin, J., Jackson, T., Fletcher, M. et al. (2011). CARMEN: Code analysis, Repository and Modeling for e-Neuroscience. *Procedia Computer Science*, *4*, 768-777.

Bechhofer, S., Buchan, I., De Roure, D, et al. (2013). Why linked data is not enough for scientists, *Future Generation Computer Systems, 29*(2), 599-611.

Borgman, C L. (2012). The conundrum of sharing research data. *Journal of the American Society for Information Science and Technology*, *63*(6), 1059-1078.

Bowden, DM., Song, E., Kosheleva, J. et al. (2012). NeuroNames: an ontology for the BrainInfo portal to neuroscience on the web. *Neuroinformatics*, *10*(1), 97-114.

Brase, J. (2009). Datacite-a global registration agency for research data. In *4th International Conference on Cooperation and Promotion of Information Resources in Science and Technology,* (pp. 257-261). IEEE.

Choudhury, S., DiLauro, T., Szalay, A. et al. (2008). Digital data preservation for scholarly publications in astronomy. *International Journal of Digital Curation*, *2*(2), 20-30.

Clark, T. (2003). Editorial: Identity and interoperability in bioinformatics. *Briefings in bioinformatics*, *4*(1), 4-6.

Collage Authoring Environment, (2013). https://collage.elsevier.com/Accessed 28th November 2013.

DataCite Metadata Working Group, (2011). DataCite Metadata Schema for the Publication and Citation of Research Data.

DataDryad, (2013) http://datadryad.org/ Accessed 28th November 2013.

Davidson, SB., & Freire, J. (2008). Provenance and scientific workflows: challenges and opportunities. In *Procs ACM SIGMOD International Conference on Management of Data* (pp. 1345-1350).

de Kamps, M. & Baier, V. (2007). Multiple Interacting Instantiations of Neuronal Dynamics (MIIND): a Library for Rapid Prototyping of Models in Cognitive Neuroscience. *Procs International Joint Conference on Neural Networks (IJCNN'2007)*, Florida

Digital Object Identifier (DOI). (2013). http://dx.doi.org/ Accessed 28th November 2013.

Eckerson, W. (1995). Three Tier Client/Server Architecture:Achieving Scalability, Performance, and Efficiency in Client Server Applications. *Open Information Systems*, *10*.

Eglen, SJ., Weeks, M., Jessop, M. et al (2014). A data repository and analysis framework for spontaneous neural activity recordings in developing retina. GigaScience 2014, 3:3 (26 March 2014).

Fielding, RT. (2000). Architectural Styles and the Design of Network-based Software Architectures, *Doctoral dissertation*. University of California, Irvine.

Freire, J., Koop, D., Santos, E. eta al. (2008). Provenance for computational tasks: A survey. *Computing in Science & Engineering*, *10*(3), 11-21.

Gardner, D., Akil, H., Ascoli, G A. eta al. (2008). The neuroscience information framework: a data and knowledge environment for neuroscience. *Neuroinformatic*s, *6*(3), 149-160.

Gibson F., Overton PG., Smulders TV. et al. (2009). Minimum Information about a Neuroscience Investigation (MINI): Electrophysiology. *Nature Precedings*. http://hdl.handle.net/10101/npre.2009.1720.2.

Goble, CA., Bhagat, J., Aleksejevs, S. et al. (2010). myExperiment: a repository and social network for the sharing of bioinformatics workflows. *Nucleic Acids Research*, *38*(suppl 2), W677-W682.

Google Web Toolkit, (2013). http://code.google.com/webtoolkit/ Accessed 28th November 2013.

Gray, J., Szalay, AS., Thakar, AR. et al. (2002). Online scientific data curation, publication and archiving (Technical Report MSR-TR-2002-74). Redmond, WA: Microsoft Research

Harris K., Henze DA., Csicsvari J. et al. (2000). Accuracy of tetrode spike separation as determined by simultaneous intracellular and extracellular measurements. *Journal of Neurophysiology*, *84*(1), 401-414.

HDF Group. (2000). Hierarchical data format version 5. *Software package,* http://www.hdfgroup.org/HDF5 Accessed 28th November 2013.

Herz, AV., Meier, R., Nawrot, MP. et al. (2008). G-Node: An integrated tool-sharing platform to support cellular and systems neurophysiology in the age of global neuroinformatics. *Neural Networks*, *21*(8), 1070-1075.

Hodge, V., Turner, A., Fletcher, M. et al. (2012). Enhancing YouShare: the online collaboration research environment for sharing data and services. *Digital Research 2012*, Oxford, UK.

Hull, D., Wolstencroft, K., Stevens, R. et al. (2006). Taverna: a tool for building and running workflows of services. *Nucleic Acids Research*, *34*(Web Server issue), 729-732.

Ingram, C., Jessop, M., Weeks, M. et al. (2012). Development of a workflow system for the CARMEN Neuroscience Portal. *5th INCF Neuroinformatics Congress*, Munich, Germany.

International Neuroinformatics Coordinating Facility (INCF) Dataspace, (2013). http://www.incf.org/resources/data-space/data-space Accessed 28th November 2013.

Jessop M., Weeks M. & Austin J. (2010). CARMEN: a practical approach to metadata management. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences, 368*(1926), 4147-59.

Kennedy, DN., Haselgrove, C., Hodge, SM. et al. (2012). CANDIShare: A Resource for Pediatric Neuroimaging Data. *Neuroinformatics*. *10*(3), 319-322.

Lakhani, KR., Jeppesen, LB., Lohse, PA. et al. (2007). The Value of Openness in Scientific Problem Solving. *Division of Research, Harvard Business School.*

Lancaster JL., Woldorff MG., Parsons LM. et al. (2000). Automated Talairach Atlas labels for functional brain mapping. *Human Brain Mapping*, *10*, 120-131.

Lawrence, B., Jones, C., Matthews, B. et al. (2011). Citation and peer review of data: Moving towards formal data publication. *International Journal of Digital Curation*, *6*(2), 4-37.

Liang, B., Fletcher, M. & Austin, J. (2010). The design and implementation of a Neurophysiology Data translation Format (NDF), 9th UK e-Science All Hands Meeting (AHM 2010), Cardiff, UK.

Littauer, R., Ram, K., Ludäscher, B. et al. (2012). Trends in Use of Scientific Workflows: Insights from a Public Repository and Recommendations for Best Practice. *International Journal of Digital Curation*, *7*(2), 92-100.

Luo, XZJ., Kennedy, D N., & Cohen, Z. (2009). Neuroimaging informatics tools and resources clearinghouse (NITRC) resource announcement. *Neuroinformatics*, *7*(1), 55-56.

Marenco, L., Ascoli, GA., Martone, ME, et al. (2008). The NIF LinkOut broker: A web resource to facilitate federated data integration using NCBI Identifiers. *Neuroinformatics*, *6*(3), 219-227.

Masud, MS. & Borisyuk, R. (2011). Statistical technique for analysing functional connectivity of multiple spike trains. *Journal of Neuroscience Methods*, *196*(1): 201-219.

McCandless, M., Hatcher, E. & Gospodnetic O. (2010). Lucene in Action, Second Edition: Covers Apache Lucene 3.0. *Manning Publications Co., Greenwich, CT, USA.*

Michener, W., Vieglais, D., Vision, T. et al. (2011). DataONE: Data Observation Network for Earth-Preserving Data and Enabling Innovation in the Biological and Environmental Sciences. *D-Lib Magazine*, *17*(1), 3.

Mittelbach, F., Goossens, M., Braams, J. et al. (2010). The LaTeX Companion. *Addison-Wesley.*

Mukhopadhyay, S., & Ray, GC. (1998). A new interpretation of nonlinear energy operator and its efficacy in spike detection. *IEEE Transactions on Biomedical Engineering*, *45*(2), 180-187.

MySQL, (2013). http://www.mysql.com/ Accessed 28th November 2013.

Néron, B., Ménager, H., Maufrais, C. et al. (2009). Mobyle: a new full web bioinformatics framework. *Bioinformatics*, *25*(22), 3005-3011.

Neurophysiology Data Translation Format (NDF) Specification – V1.2.1. (2013) http://www.carmen.org.uk/standards/CarmenDataSpecs.pdf Accessed 28th November 2013.

Nowakowski, P. Ciepiela, E., Harę lak, D. et al. (2011). The Collage Authoring Environment. *Procedia Computer Science*, *4*, 608-617.

Oinn, T., Addis, M., Ferris, J. et al. (2004). Taverna: a tool for the composition and enactment of bioinformatics workflows. *Bioinformatics*, *20*(17), 3045-3054.

Quiroga, RQ., Nadasdy, Z., & Ben-Shaul, Y. (2004). Unsupervised spike detection and sorting with wavelets and superparamagnetic clustering. *Neural Computation*, *16*(8), 1661-1687.

Reich M., Liefeld T., Gould J. et al. (2006). GenePattern 2.0. *Nature Genetics, 38*(5), 500-501.

ResearcherID, (2013). http://www.researcherid.com/ Accessed 28th November 2013.

Ritz, R., Kotaleski, JH., Strandberg, P. et al. (2008). A new software center for the neuroinformatics community. *BMC Neuroscience*, *9*(Suppl 1), P89.

Romano, P., Bartocci, E., Bertolini, G. et al. (2007). Biowep: a workflow enactment portal for bioinformatics applications. *BMC bioinformatics*, *8*(Suppl 1), S19.

Run My Code. (2013). www.runmycode.org/ Accessed 28th November 2013.

Sernagor, E. (2011). CARMEN for the storage and analysis of rich datasets obtained from 60 and 4,096 channels MEA recordings of retinal activity (2011). *CARMEN Consortium Meeting*, Newcastle University, UK.

Shahid, S., Walker, J. & Smith, LS. (2010). A new spike detection algorithm for extracellular neural recordings, *IEEE Transactions on Biomedical Engineering*, *57*(4), 853-866.

Simonotto, J., Kaiser, M., Sernagor, E. et al. (2011). Network Extraction and Analysis in CARMEN (2011). *CARMEN Consortium Meeting*, Newcastle University, UK.

Simonotto, J., Eglen, S., Kaiser, M. et al. (2009). Analysis of spontaneous activity patterns in the developing retina: extracting and analyzing dynamical networks. *Society for Neuroscience (SfN) Meeting, Chicago, USA.*

Singh, J. FigShare (2011). *Journal of Pharmacology & Pharmacotherapeutics*, *2*(2), 138.

Sivagnanam, S., Majumdar, A., Yoshimoto, K. et al. (2013). Introducing The Neuroscience Gateway. *Proceedings of the 5th International Workshop on Science Gateways*, Zurich, Switzerland.

Springer (2014). http://www.springeropen.com/about/supportingdata Accessed 28th April 2014

Süptitz, T., Weis, SJJ. & Eymann, T. Was müssen Virtual Research Environments leisten?-Ein Literaturreview zu den funktionalen und nichtfunktionalen Anforderungen. (2013). In *Wirtschaftsinformatik* (p. 21).

Teeters JL., Harris KD., Millman KJ. et al. (2008). Data sharing for computational neuroscience. *Neuroinformatics*, *6*(1), 47-55.

Weeks, M., Jessop, M., Fletcher, M. et al. The CARMEN software as a service infrastructure. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences, 371*(1983), 2013.

White Rose Grid. (2013). http://www.wrgrid.org.uk/ Accessed 28th November 2013.

Woodman, S., Hiden, H., Watson, P. et al. (2009). Workflows and Applications in e-Science Central. *All Hands Meeting*, Oxford, UK.

Yarmey, L. & Baker, KS. (2013). Towards Standardization: A Participatory Framework for Scientific Standard-Making. *International Journal of Digital Curation*, *8*(1), 157-172.

Zou C., Ladroue C., Guo S. et al. (2010). Identifying interactions in the time and frequency domains in local and global networks - A Granger Causality Approach. *BMC Bioinformatics*, *11*, 337.