HBP Visualisation Software Catalogue (D7.4.1 – SGA2)

Project Number:	785907	Project Title:	Human Brain Project SGA2

Document Title:	HBP Visualisation Software Catalogue
Document Filename:	D7.4.1 (D45.1 D84) SGA2 M24 ACCEPTED 200731.docx
Deliverable Number:	SGA2 D7.4.1 (D45.1, D84)
Deliverable Type:	Websites, patents filling, etc.
Work Packages:	WP7.4
Dissemination Level:	PU = Public
Planned Delivery Date:	SGA2 M24 / 31 Mar 2020
Actual Delivery Date:	SGA2 M24 / 10 Mar 2020; Accepted: 31 Jul 2020
Author(s):	Benjamin WEYERS, UT (P130) Nico FELD, UT (P130) Daniel ZIELASKO, UT (P130)
Compiled by:	Benjamin WEYERS, UT (P130)
Contributor(s):	
SciTechCoord Review:	Martin TELEFONT, EPFL (P1)
Editorial Review:	Annemieke MICHELS, EPFL (P1)
Description in GA:	The HBP Visualisation Catalogue gives an overview of all visualisation tools and methods available in the HBP. It is a publicly available website that is also integrated into the Collaboratory. It contains semantic query functionality to support users in finding the right visualisation method and software for their specific needs.
Abstract:	This document describes the final version of the HBP Visualisation Catalogue, which has been transformed into an extension of the meta data set searchable and usable in the EBRAINS Knowledge Graph. This document describes in detail the developed and implemented meta data schema, the implemented curation process, as well as the actual integration into the knowledge search engine.
Keywords:	software meta data, software features, knowledge graph, meta data curation
Target Users/Readers:	neuroscience community, computer scientists, consortium members

Table of Contents

Table of Tables

Table of Figures

1. Introduction

Originally, we planned to develop a "…visualisation software catalogue […|that] implement[s] a semantic query functionality to support users in finding the right visualisation method and software for their specific needs." During SGA2, it became clear that not only offering access to meta data on visualisation software is needed, but that general software meta data are needed as well. Therefore, we slightly changed our activity from preparing a visualisation software catalogue to a software meta data catalogue. As planned, this activity was closely coordinated with collaborators of PLUS, the Collaboratory, the Knowledge Graph (KG), and with the data curation teams (please refer to D7.6.1 and D7.6.2 for more details). We established curation processes for the step-wise research and integration of Software Metadata into the KG from various sources, in close collaboration with support by the Data Curation Team and Andrew DAVISON (CNRS). All processes are documented in the catalogue's GitHub Repository (https://github.com/bweyers/HBPVisCatalogue) and described in detail in this Deliverable. The curation processes are transparently documented in the GitHub repository's issue tracker: https://github.com/bweyers/HBPVisCatalogue/issues.

We started to integrate meta data on software from the Collaboratory software catalogue, the High Performance Analytics and Computing (HPAC) Guidebook, as well as suggestions from Consortium members. The catalogue is planned to expand beyond only capturing HBP-developed software: it will also include general software, relevant for neuroscience and/or provided on EBRAINS resources.

Table 1 provides information on the actual catalogue. The final result of the curation process can be accessed through the HBP Knowledge Graph search front end here:

https://kg.ebrains.eu/search/?facet_type%5b0%5d=Software&facet_type[0]=Software.

Table 1: List of relevant links

Component	Link to	URL
C2707	Curation Repository	https://github.com/bweyers/HBPVisCatalogue
	Technical Documentation	https://github.com/bweyers/HBPVisCatalogue/wiki
	User Documentation	https://github.com/bweyers/HBPVisCatalogue/wiki

1.1 Impact for the HBP

Software metadata is now searchable in the EBRAINS Knowledge Graph and can be used as entities in terms of advanced semantic descriptions, e.g. in terms of provenance tracking, in relation with curated data sets, as well as for making software searchable from the outside of the HBP (KG search interface is openly accessible). The original goals of making visualisation software tools available and semantically searchable have been achieved, due to the use of the established software stack provided by the EBRAINS Knowledge Graph, and this has been extended towards general software as well. Additionally, the meta data-gathering process is supported by a transparent meta data curation process.

The impact for the HBP has been broadened from a strong focus on visualisation software to a software collection giving access to tools relevant for neuroscience.

1.2 Contributors

The following list contains only names of the representatives of the various groups involved in the conceptualisation and development of the software meta data catalogue:

UT, P130 – Nico FELD, Daniel ZIELASKO, Sascha FELD, Annika HARTWICH, Daniel ZEILER, Benjamin WEYERS (main developer and SW meta data curators)

Data Curation Teams – Maja AMEDJKOUH PUCHADES (UIO, P81), Lyuba ZEHL (JUELICH, P20), Andrew DAVISON (CNRS, P10), and others

Knowledge Graph Team – Oliver SCHMID (EPFL, P1)

Collaboratory Development & PLUS Team – Allan FRANCANI, Marc MORGAN (EPFL, P1)

1.3 Overview of this document

The goal of this document is to highlight all relevant aspects (from structure over curation to use) of the software meta data accessible through the EBRAINS Knowledge Graph. Section 2 will introduce i) the metadata schema that has been developed (Section 2.1); ii) the implemented curation process, used to integrate software metadata into the Knowledge Graph in a controlled way (Section 2.2); and iii) the search front end provided by the EBRAINS Knowledge Graph search UI (Section 2.3).

2. Software Metadata – Schema, Curation, Search

Software meta data are described following a meta data schema that will be introduced in Section 2.1. To be able to enter more than one version of a software, software meta data are organised by means of software projects, as illustrated in Figure 1. This is specifically needed for handling provenance information entering the Knowledge Graph. Besides, each software may be related to software features, which are the major search criteria to search for a specific tool addressing a specific problem.

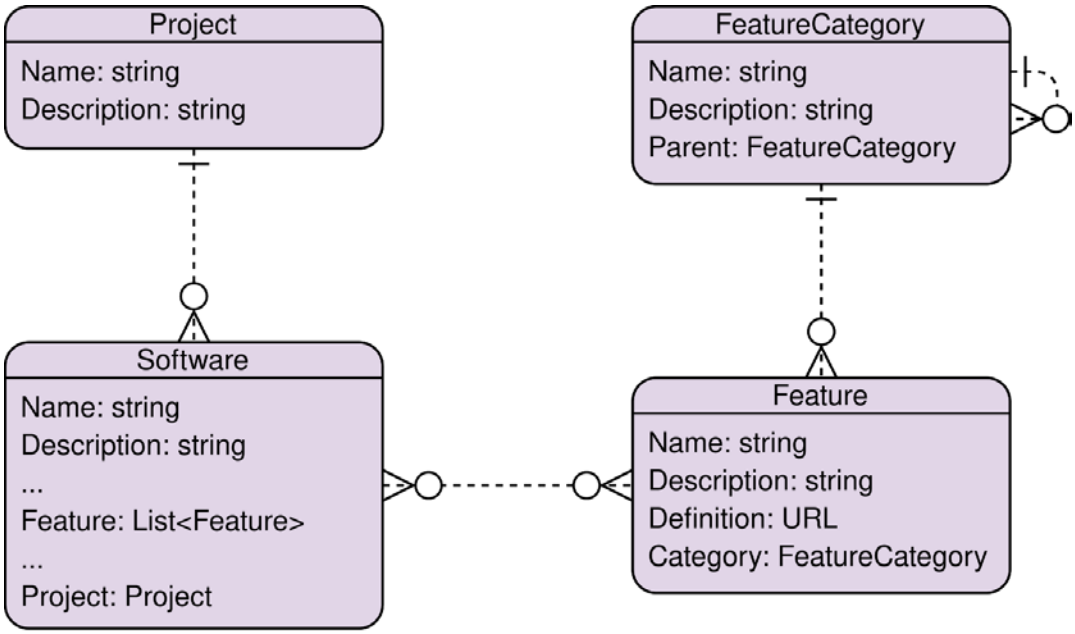

Figure 1: Structure of entities in the Knowledge Graph

Structure of entities in the Knowledge Graph to enable handling of multiple versions of a software.

2.1 Software Metadata Schema

For standardised curation of meta data as well as for making meta data sharable and reusable, during SGA2 a suitable software meta data schema has been developed in close cooperation with Andrew DAVISON (CNRS) and by discussions with the HBP data curation team. The development closely focused on existing schemas (schema.org, Collaboratory SW catalogue, NeuroShapes) to keep a certain level of compatibility to existing schemas. However, not all existing schemas / elements of schemas were suitable for the purpose envisioned for meta data on software in the EBRAINS Knowledge Graph. In an iterative process, the final schema has been developed and is presented in detail below.

2.1.1 Meta Data Schema for Software (Status Feb. 2020, implemented in the Knowledge Graph)

Table 2 shows all fields of the meta data schema. All fields in **red** are considered mandatory entries for the schema. The rest is optional and, thus, is potentially missing in entries in the EBRAINS Knowledge Graph. When possible, all used field values (e.g. software licenses) refer to WikiData in addition to back up term definitions by external Knowledge Graphs.

Table 2: Meta Data Schema for Software as implemented in the EBRAINS Knowledge Graph

Name	Name of the software including the version number.
Summary	A short summary describing the software.
Description	A more detailed description of the software.
Created at	The date on which the described version has been created (e.g. published, or date of release)
Application category	One of: Application, Library, Plugin, Module
Operating System	The operating system the software can be / is deployed on.
Software requirements	Software requirements for installing & using the Software. Not for development/building. requirements.txt of python project must not be included.
Homepage	URL pointing the software's home page, e.g. a GitHub page.
License	License under which the software is released / available. Use None if there is no license available. Add version number of license if available.
Funder	Organisations provided funding for development of the software, e.g. the HBP
Keywords	Keywords describing the software, e.g. neural simulator
Version	Version number of the software. If none provided, use the date this meta data entry has been created in the sense of a "last visited" reference in scientific publications.
Language	Language used in the software (not programming language)
Citation	Citations to publications describing the software
Release notes	Release notes of the described version of the software
Screenshot(s)	URLs to screenshots of the software
Author(s)	List of authors (e.g. main developers) of the software
Copyright holder(s)	Copyright holders
Copyright year	Copyright year
Documentation	URLs to web pages holding the software's documentation
Help	URLs to location providing help, e.g. issue trackers or mailing lists
Source Code	URL to the source code
Programming language	Programming language(s) used
Components	Software, which is part of this software and also exists as software entity in the KG

Is accessible for free	Yes or No
Icon	URL to an icon of the software
Device	One or more of: Desktop, Mobile, Web, Server, HPC
Features	Please find more details on Features in the next section below
Input Formats	List of names of the File-Format including its usual ending, e.g. NeuroML(XML))
Output Formats	List of names of the File-Format including its usual ending, e.g. NeuroML(XML))

2.1.2 Software Features

A main focus in the curation process is the identification of characteristic software features to support the user to search for specific software tools, with certain characteristics. The identified set of software features may change during the entry of further software tools into the KG as the set of features is not assumed to be complete but may converge over time. The list below provides the current status (Feb. 2020).

- **Data Types:** includes all Data Types a software can read, process, output, etc.
  - **Abstract:** includes all Data Types, which are an abstract representation of certain data.

    Graph Data Types: describes any form of graphs, e.g. neural networks.

    Statistical Data Types: describes any statistical data, e.g. results of a survey/study.

    Time Series Data Types: describes data distributed over a specific time, e.g EKG data.

    Positional Data Types: describes various positions in an arbitrary space, e.g. landmark locations.

    Metadata Data Types: describes any form of meta data, e.g. a catalogue of various software.
  - **2D Image Data:** includes all Data Types, which are any 2D image representation.

    Raster Image Data Types: describes images, which are represented in a raster format, e.g. 2D MRI data.

    Vector Image Data Types: describes images, which are represented in a vector format, e.g. SVG-images.
  - **3D Scientific Data:** includes all Data Types, which are scientific data in 3D-Space.

    3D-Scalar Data Types: describes scalar data, located in a 3D Space, e.g. 3D MRI data.

    3D-Vector Data Types: describes vector data, located in a 3D Space, e.g. 3D PLI data.

    Tensor Data Types: describes tensor data, located in a 3D Space.
  - **3D Geometry Data:** describes 3D geometry data, e.g. a 3D mesh.
- **Usage:** includes possible tasks the software can be used for.
  - **Interactive Analysis:** allows the user to interactively analyse given data.
  - **Presentation/Visualization:** allows the user to prepare given data for presentation and/or visualization.
  - **Data Processing:** allows the user to process given data into another form.
  - **Simulation:** allows the user to run simulations on given data.
  - **Provenance:** allows the user to run or create provenance tracking.

  - **Profiling:** allows the user to do profiling of other software.
  - **Modelling:** allows the user to create or edit models such as 3D Geometry Data, Neural Networks, etc.
- **Display:** includes various display devices the software supports.
  - **Desktop Environment:** the software can be used in a normal desktop environment.
  - **Virtual Reality:** the software can be used in a virtual reality environment.
  - **Augmented Reality:** the software can be used in an augmented reality environment.
  - **Tiled Display Wall:** the software can be used with a tiled display wall.
- **User Interface:** Interfaces enabling the user to interact with the software
  - **Graphical User Interface:** the software does have a Graphical User Interface, e.g. offering widgets.
  - **Commandline Interface:** the software does have a CLI which can be used, e.g. with terminals and/or shell scripts-shell is executing software.
  - **Scripting Interface:** the software can process scripts provided as files in specific format-software is processing (executing) script.
- **System Architecture:** _includes various system architectures the software supports
  - **Heterogeneous Architecture:** the software can be build, deployed, and/or used on systems combining hardware of various types for processing, e.g. CPU and GPU
- **Development:** includes various features of the software if it can be used for developing other software.
  - **Parallel Programming:** the software supports parallel programming.

2.2 Curation Process

Curation has been established to handle and to pipeline entering software meta data into the EBRAINS Knowledge Graph. Depending on how software has been identified to be entered into the Knowledge Graph, various types of processes have been implemented as detailed below. In general, data have been gathered in a first step followed by the actual curation, which is similar for all types of data gathering processes.

Used abbreviations below: Software Meta Data Curation Team – ST, Data Curation Team – DT

2.2.1 Data Gathering Process

2.2.1.1 Internal Meta Data Gathering Process

1) Each member of ST gets a copy of this schema template and creates a new tab for each software-entity which he/she curates (in Google Spreadsheet)
   a) SW tools are gathered from the following sources:
      HBP Collaboratory SW Catalogue
      HPAC Guidebook
      PLUS
      NIRTC Data Base (https://www.nitrc.org/)
2) Each tab gets pre-filled with all known information by ST member through web research
3) **Enter Curation Process**

2.2.1.2 Trigger by DT

1) DT member sends SW name & version to ST by opening an issue
2) DT member OR ST member prefills schema using a **copy** of the schema template
3) **Enter Curation Process**

2.2.1.3 All others (external users)

1) External user downloads the schema template from the GitHub page (https://github.com/bweyers/HBPVisCatalogue/wiki/Templates)
2) External user prefills template with all information they can provide
3) External user creates issue (on GitHub https://github.com/bweyers/HBPVisCatalogue/issues) and attaches the prefilled spreadsheet
4) **Enter Curation Process**

2.2.2 Curation Process

1) ST gets notified by issue tracker & ST member gets allocated to the issue
2) In a curation meeting, ST lead adds *TODO*'s into the comments
3) The corresponding ST member processes the *TODO*'s
4) ST member asks developers/responsible PI of corresponding software to add missing information or point us to it (optional if data is missing)
5) If all required fields are filled, ST member transfers software-entity into the KG with the KG-Editor.
   a) "Link to external definition" will be filled with a link of an external knowledge base, like https://www.wikidata.org/
6) Developer gets final confirmation from the responsible ST member
7) Release

2.2.3 Workflow Implementation

For handling the curation process, ST uses the GitHub issue tracker (https://github.com/bweyers/HBPVisCatalogue/issues). Labels are used to indicate the status of the meta data entity in the process. This makes the process transparent, not only to the ST, but also to the outside world. The labels are read as follows:

- All issues addressing the curation of software entities / meta data are marked with the *Software* label
- ***ongoing***: We are working on gathering meta data
- ***in curation***: ST lead has to look at it, check the content and either send it back to ST members (mark it as ongoing again) or sent it for revision to the developer
- ***send for revision***: Authors of a software have received an email asking to check the google sheets/the meta data we found
- ***to be integrated***: data now has to move into the KG
- ***to be released***: the meta data can be released to be publicly visible

- ***new***: this software was suggested by others and an ST member will start gathering meta data soon.
- ***attention needed***: this software has issues, which must be discussed in a meeting and/or with a contact person / developer

When published in the KG, issues will be closed. Thus, all finished software entries can be found under closed issues. After integration, the filled out template will be archived in a private google-sheet.

2.3 KG Integration and Search

Entered meta data entries on software are searchable (after release) in the EBRAINS Knowledge Graph search front end. Figure 2 shows the search mask, whereas Figure 3 shows a so called *software meta data card*.

Currently, only a search for the name of a software and a filter for the supported operating system is provided. In the near future, we plan to support filters for at least:

- Application Category
- Operating System
- Language
- Programming Language
- Is accessible for free
- Device
- Features
- Input/Output Formats

The software card then also should include at least:

- Authors
- Software requirements
- Keywords
- Language
- Citations
- Release notes
- Programming Language
- Components
- Is accessible for free
- Icon
- Device
- Input/Output formats

The further implementation and visualisation will be developed in cooperation with Oliver SCHMID (EPFL, P1).

Currently (26 Feb 2020), 15 software tools are searchable, 27 are in a late stage “>=in revision” of the curation process, 18 just entered the process “<in revision”. We plan to release 42 entities in total by the end of SGA2.

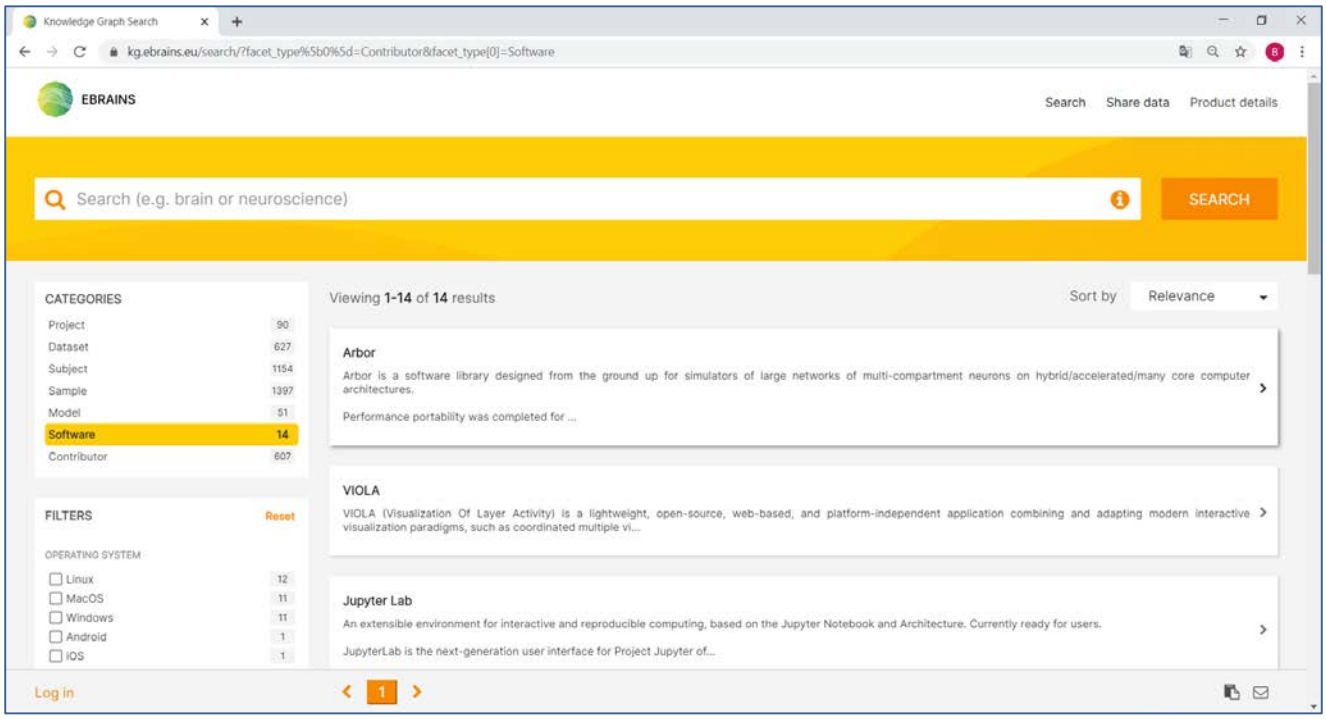

Figure 2: Search UI for the EBRAINS Knowledge Graph

Screenshot of search UI for the EBRAINS Knowledge Graph showing a list of all searchable software meta data entities (28 Feb 2020).

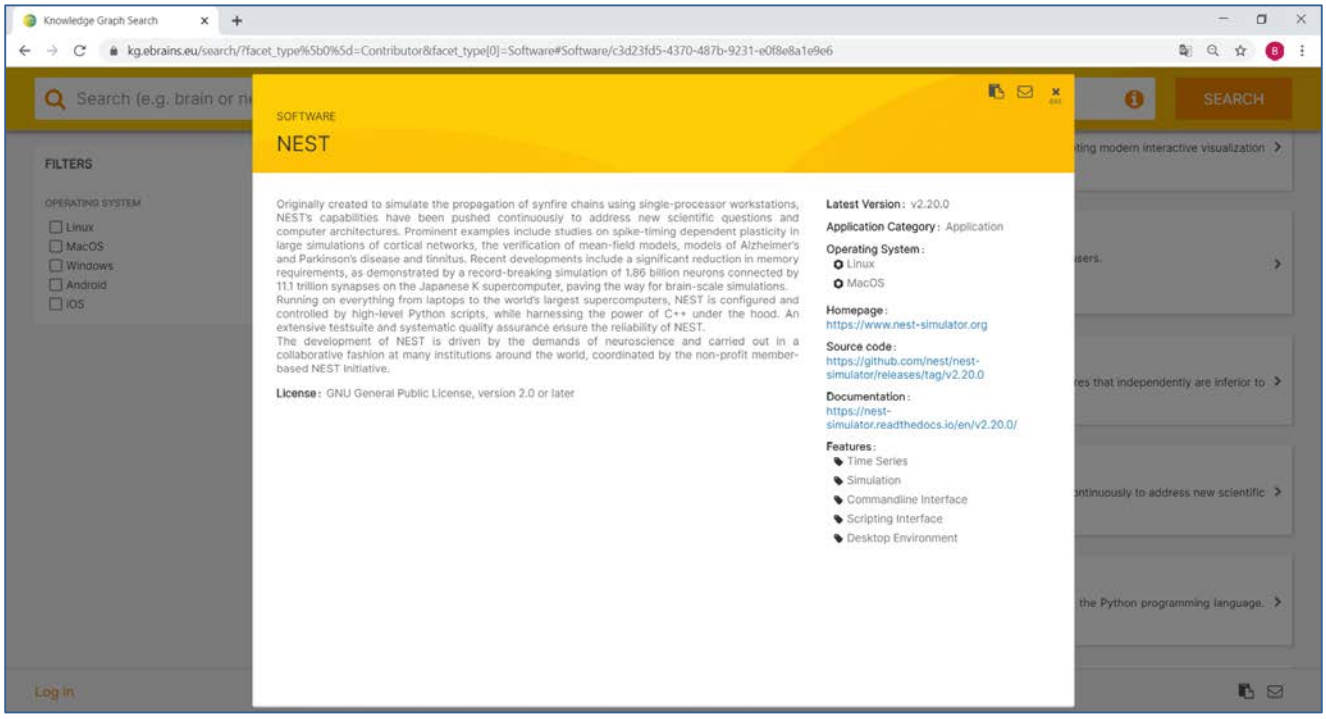

Figure 3: Software meta data card.

Screenshot of software meta data card in its current version (27 Feb 2020).