D5.1 EBRAINS Closed-loop AI and robotics workflows (D5.1 - SGA3)

Neurorobotics Platform release with improved functionality, extended content, and updated inventory of related closed-loop models, tools, and services prepared for, or released through the EBRAINS portal.

Figure 1: Whiskeye robot explores a virtual environment on the NRP through tactile sensory input

Controlled by a complex custom cognitive architecture, a Whiskeye robot explores a virtual environment on the NRP through tactile sensory input provided by artificial whiskers. This experiment is now available in release 3.1 of the NRP, enabling interested users to reproduce results, modify the cognitive architecture and compare its performance to the reported baseline, etc. (see Section 3)

Project Number:	945539	Project Title:	HBP SGA3

Document Title:	EBRAINS Closed-loop AI and robotics workflows (D5.1)
Document Filename:	D5.1 (D48) SGA3 M9 ACCEPTED 2100504.docx
Deliverable Number:	SGA3 D5.1 (D48)
Deliverable Type:	Other: Software
Dissemination Level:	PU = Public
Planned Delivery Date:	SGA3 M9 / 31 Dec 2020
Actual Delivery Date:	SGA3 M10 / 12 Jan 2021; resubmitted 31 Mar 2021; accepted 04 May 2021
Author(s):	Fabrice MORIN, TUM(P56)
Compiled by:	
Contributor(s):	
WP QC Review:	Victoria NEUMANN, TUM (P56);
WP Leader / Deputy Leader Sign Off:	Fabrice MORIN, TUM (P56)
T7.4 QC Review:	Martin TELEFONT, EPFL (P1) – sci/tech QC Annemieke MICHELS, EPFL (P1) – editorial QC
Description in GA:	Neurorobotics Platform release with improved functionality, extended content, and updated inventory of related closed-loop models, tools, and services prepared for, or released through the EBRAINS portal.
Abstract:	This document provides a high-level overview of the evolution of the Neurorobotics Platform (NRP) in SGA3, focusing on the update of its software stack from v3.0 to v3.1. This latest version is more functional, more usable, better adapted to the demands of high-performance computing. Of particular importance was the update to Python 3.0. The NRP Code is open source and available via Bitbucket repository. The use of the NRP in learning experiments has been facilitated, which will be fundamental for implementing learning/optimisation services. In addition, an experiment by the University of Bristol was included in the 3.1 release of the NRP. Moreover, the evolution of the philosophy of the platform and the changes that users can expect to discover in 2021 is addressed. V4.0 is on the horizon and will bring significant changes and new services. For example, large-scale spiking neural networks in NEST are about to become available to NRP users. We are likewise putting a focus on modularity which will provide users with the interfaces and APIs to connect simulation engines. The undertaken and planned efforts are contributing to a future-proof NRP and, with that, considerably enhance the maintainability of its codebase, which is also essential for the integration of the NRP into EBRAINS.
Keywords:	neurorobotics, embodiedsimulation, closedloop, neuroscience, embodiedAI
Target Users/Readers:	Computational neuroscience community, computer scientists, Consortium members, embodied AI researchers, roboticists, general public, neuroscientific community, Platform users, students.

Table of Contents

Table of Tables

Table of Figures

History of Changes made to this Deliverable (post Submission)

Date	Change Requested / Change Made / Other Action
10 Jan 2021	Deliverable submitted to EC
04 Mar 2021	Resubmission with specified changes requested in Review Report Main changes requested: • Change 1: Future activity/deliveries should be indicated, p.17
30 Mar 2021	Revised draft sent by SC4 to PCO. Main changes made, with indication where each change was made: • Change 1: addressed in Section 6. Annex
31 Mar 2021	Revised version resubmitted to EC by PCO via SyGMa

1. Introduction

The present document provides a high-level view of the evolution of the Neurorobotics Platform (NRP) since April 2020, focusing on the update of its software stack (Section 2), the supporting concrete use cases as a basis for future EBRAINS services (Section 3), as well as the evolution of the very philosophy of the platform and the changes that users can expect to discover in 2021 (section 4).

2. Updating the NRP software stack

After the release of the NRP v3.0 in April 2020, a decision was made to limit the number of new features to be integrated into release 3.1, and to focus instead on a full update of the software stack (see Table 1 below). This decision was motivated by multiple factors.

First, supporting Python 3 in the NRP was becoming essential insofar as the Python Software Foundation discontinued support for the 2.x branch of this programming language on the 1st of January 2020. Second, NRP users cannot be expected to work with out-of-date versions of their operating system and software tools: this is a fundamental obstacle to the adoption of the NRP across all communities. Finally, the complex network of dependencies introduced by the NRP software stack made it particularly difficult to maintain and expand a code base comprising multiple deprecated components.

At a time when integration into the EBRAINS infrastructure and creation of production-grade services becomes a priority in the HBP, this aspect could no longer be ignored. It was, therefore simply not possible to further postpone the update of the NRP software stack. Table 1 below provides a summary of the evolution of the latter between NRP releases 3.0 and 3.1.

Table 1: Comparison of the main components of the software stack for NRP releases since April 2020

Release 3.0	Release 3.1
Python 2.7	Python 3.8
Ubuntu 18.04	Ubuntu 20.04
ROS Melodic	ROS Noetic
Gazebo 9	Gazebo 11
NEST 2.12	NEST 2.18
PyNN 0.9.4	PyNN 0.9.5

The source code of the NRP is open source and available on the following Bitbucket repository: https://bitbucket.org/hbpneurorobotics/neurorobotics-platform/src/master.

The Docker installer for local install is available at https://neurorobotics.net/local_install.html.

The NRP documentation is available at https://neurorobotics.net/Documentation/nrp/index.html.

3. From NRP tools to NRP services

While this updating effort mobilised significant resources of the NRP development team, a series of efforts were undertaken to make the NRP future-proof (see Section 4) and to better support current users' needs as well as services based on the platform.

For example, the Virtual Coach was made stand-alone (installable through apt-get) so as to facilitate the use of the NRP in learning experiments. Users now can orchestrate batch execution of NRP experiments without the need for a full "dummy" installation of the NRP just to access the Virtual Coach. Usability is thus significantly improved in workflows relying on distributed execution of parallel experimental instances. This will prove fundamental for implementing learning/optimisation services (e.g. L2L, IMPALA, etc.).

Another example of developing a service based on the online NRP consists in the inclusion of an experiment by the University of Bristol in release 3.1 of the NRP, with the specific purpose of using this online experiment for a publication (currently in preparation). Controlled by a complex custom cognitive architecture – entirely accessible by the users – a Whiskeye robot explores a virtual environment on the NRP through tactile sensory input provided by artificial whiskers. Having this experiment available on the online version of the NRP enables scientists to publish papers based thereon with a direct link to their work. Interested readers can then access the experiment in a few seconds, reproduce results contained in the paper, modify the cognitive architecture and compare its performance to the reported baseline, etc. This approach to sharing scientific results is bound to become the norm, and the NRP demonstrates here its relevance for multiple communities (computational neuroscience, robotics, etc.).

4. Working for the future: preparing the 2021 release of NRP 4.0

Many efforts undertaken since April 2020 are not translated yet into the latest 3.1 NRP release. However, they are essential to the evolution of the NRP and are very much indicative of the many new functionalities that users can expect to see in 2021. V4.0 is indeed on the horizon and will bring a host of major changes and new services.

First, large-scale spiking neural networks in NEST are about to become available to NRP users. Enabled by a new client-server communication architecture between NEST 3.0 (https://nest-simulator.readthedocs.io/en/latest/guides/nest2_to_nest3/nest2_to_nest3_overview.html) and the closed-loop engine (CLE) of the NRP, large-scale spiking neural networks will be run inside embodied NRP simulations by leveraging the compute power of the Piz Daint cluster at the Swiss National Supercomputing Centre in Lugano (CSCS, Switzerland). Current prototypes already comprise more than a 1M neurons and 1B synapses, and scaling-up these figures by at least one order of magnitude are expected in the coming months. 2021 will thus see the establishment of a proper embodied simulation service for large-scale spiking networks based on the technical basis laid out by these prototypes.

Another essential change that will come for the NRP in 2021 is modularity: we aim to provide users with the interfaces and APIs to connect simulation engines that so far have remained beyond the reach of the NRP. To this end, the CLE of the NRP is being entirely re-written in C++20, save for the transfer functions that will remain in Python for enabling users to modify them interactively. The client-server architecture successfully used with NEST is being generalised to all other simulators currently in use with the NRP (including Gazebo, which will become optional where it is currently a core component of the NRP), and new interfaces are being established (most of them will follow the gRPC standard). Soon, NRP users will be able to orchestrate the execution of any number of simulation engines through the CLE and ensure fully synchronous communications between those engines for fully deterministic scientific simulations.

To support this modularity through the graphical user interface of the NRP, a complete refactoring of the NRP front end was decided. This came at a time when support for the underlying technology (AngularJS) of the front end was discontinued and made this significant investment of resources as unavoidable as desirable. The open-source React framework was chosen for its capability to support modularity through the embedding of microservices / micro-front ends. This will enable both NRP software developers and users to evolve the NRP front end in order to suit their needs for the foreseeable future.

Finally, the NRP development team made considerable strides towards the establishment of a continuous integration / continuous deployment (CI/CD) process. While at first glance, this may seem irrelevant to NRP end users, it will actually become an essential tool to deploy custom versions of the NRP. As such, it may soon support a new type of service, whereby custom Docker images (with user-supplied custom simulation engines, cognitive architectures, Virtual Coach scripts, etc.) will be deployed online for teaching, complex learning workflows, etc.

5. Conclusion

The NRP keeps getting better: release 3.1 is more functional, more usable, better adapted to the demands of high-performance computing. It evolves towards a modular architecture through the ongoing refactoring of both its front-end and back-end components. This will make the NRP future-proof and greatly enhances the maintainability of its codebase, which are essential characteristics in the perspective of the integration of the platform into the EBRAINS infrastructure. Finally, it has initiated the transition from prototype experiments to user-facing high-quality services. 2021 is shaping up to be a special year for NRP users.

6. Annex: Roadmap of upcoming activities

The following is a tentative timeline of the software development activities of the NRP, provided for information purpose only. It does not contain information of relevance to the NRP users about the current functionalities of the NRP but is meant to provide more concrete information regarding what they can expect in the coming months and years.

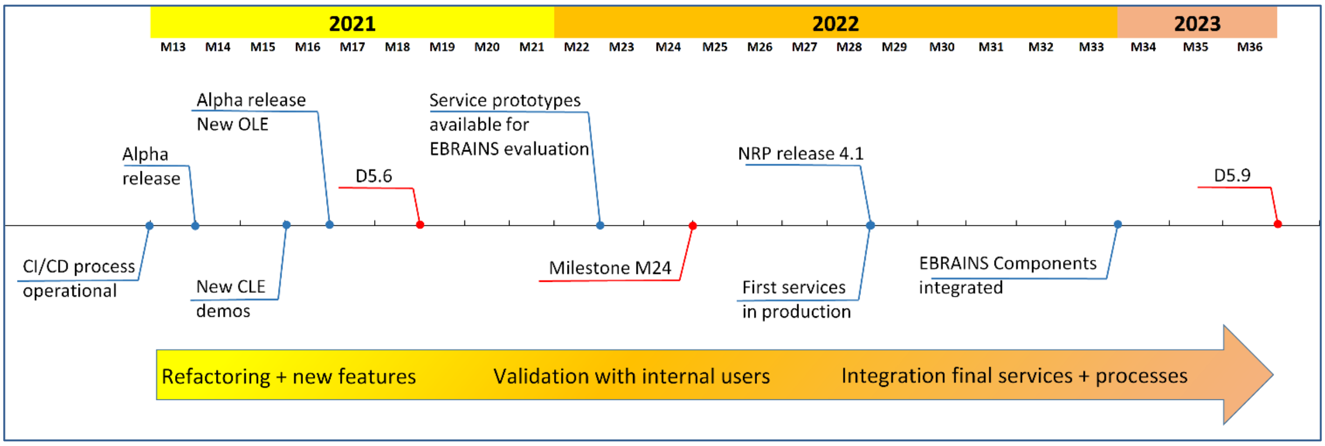

Figure 2: Tentative timeline of software development activities of the NRP.

The numbered months are related to the timeline of the Human Brain Project. Blue lines indicate internal deadlines set by the software development team of the NRP, red lines indicate Deliverables and Milestones included in the Description of Action of the Human Brain Project of relevance to the NRP users. Details for each element are provided in the text.

6.1 Significant activities and deliveries

The various action items contained in Figure 2 are described below. The months of delivery are given in reference to the timeframe of the Human Brain Project:

- M12: Deployment of Continuous Integration / Continuous Delivery (CI/CD) process: while this is process is transparent to the users, it is absolutely essential to ensure quality of delivered software and represents a significant milestone in the software development process of the NRP.
- M13: Alpha release of new CLE as standalone
  - Alpha version of the API tested
  - Documentation extensive and available on website
  - Shared with selected partners inside HBP (e.g. Charité Berlin, FZ Jülich)
- M15: Demonstrators based on new CLE (alpha version) showcasing integration of new simulation engines shared with HBP users:
  - NEST + Gazebo: to be tested and validated both locally and on Piz Daint cluster at CSCS.
  - OpenSim: tested locally with arm26 model (https://github.com/opensim-org/opensim-models/tree/master/Models/Arm26) controlled through a mechanism yet to be decided (with EPFL).
  - OpenAI Gym as a CLE "engine" (although the terminology does not suit OpenAi Gym, that is how the CLE will interface with that framework).
- M16: Alpha release of new Open-Loop Engine (OLE)[1] in standalone version

[1] The OLE is a simplified version of the CLE where constraints related to synchronous communication are removed. As such, the OLE is specifically geared towards robotics applications and real-time multi-engine execution with asynchronous communications.

  - API tested
  - Documentation available
  - Shared with selected partners inside HBP (e.g. KTH, Uni. of Sheffield)
- M18: HBP Deliverable D5.6 (D53): Public report detailing the status of preparation of EBRAINS services, as well as the status of NRP v4.0 in local installation. This v4.0 will be based on the new CLE and will feature the new REACT-based front-end of the NRP.
- M22: Two service prototypes (see section 1.2) available for evaluation by EBRAINS.
- HBP Milestone M24: Provision of integrated dynamical closed-loop models developed within the HBP, useable on the NRP.
- M28: Release of NRP v4.1, including new OLE and related demonstrators, as well as consolidated client-server interfaces between the CLE and a set (to be confirmed) of simulations engines (OpenSim and, e.g. PyBullet, TVB, etc.).
- M28: First two NRP-based services moving into production (see section 1.2), supported by apposite GUIs for NRP-based services on HPC infrastructure, and compatible with all versions of the NRP up to 4.1
- M33: Complete integration into the NRP of all EBRAINS-level components:
  - EBRAINS federated user resource management system operational.
  - Data provenance tracking operational.
  - User roles properly inherited from EBRAINS Collab.
  - CI/CD process fully operational and in line with final EBRAINS specifications.
  - Repositories containing NRP code fully migrated in line with EBRAINS specifications.
  - All quality control procedures specified by EBRAINS passed.
- M36: HBP Deliverable D5.9 (D56): Public report detailing the latest features available on the NRP and the full list of EBRAINS services based thereupon.

6.2 Description of the suggested EBRAINS services based on the NRP

The NRP itself can be considered both as a tool and a service (which has been online for many years). However, we intend to provide more value to our users by offering additional services based on the NRP with specific purposes. The first two of such new EBRAINS services will thus be (pending approval by EBRAINS management): 1) the NeuroRobotics Public Library (NRPL); and 2) embodied simulations of very large-scale spiking neural networks in NEST.

The NRPL is to be a service offered to both the Robotics and Neuroscience communities, intended to run on an Openstack park of Virtual Machines (e.g. the Pollux cluster at CSCS). The NRP Public Library is a service where experiments can be shared by an initial “owner” / author, and offered to NRP users through a graphical web interface. The NRP Public Library is intended to store the experiment files and (where necessary) actual datasets produced by the corresponding experiment in long-term archival storage. From their browser, the users should be able to browse these experiments, clone experiment files to their private space, and finally run and modify them. This service is intended to support lectures in the academic world and to offer both code and simulation engines as supporting information for scientific publications. While this functionality is closely related to currently available functions of the online NRP (e.g. cloning and sharing experiments), it needs to go beyond as there must be mechanisms for: 1) keeping the saved experiments and data safe (from, e.g. hard drive failure) by leveraging an EBRAINS solution for long-term safe data storage (e.g. SWIFT storage); 2) handling the evolution of the NRP itself (e.g. versioning and interactive deployment). Indeed, such a publishing service cannot be sustainable if the experiments available online are broken by updates to the NRP itself. This entails keeping a copy of every Docker image deployed online across time (e.g. in the EBRAINS Docker registry, Harbor), but also making sure that the Openstack infrastructure (i.e. Virtual Machines) remains compatible with every one of those images.

Provision of embodied simulations of very large-scale spiking neural networks in NEST is a service intended for the Neuroscience and Neuromorphic Computing communities. Intended to run on the Piz Daint cluster at CSCS for the NRP back end and simulation engines, and with a virtual machine (e.g. on Pollux at CSCS) running the NRP front end, this service addresses the need of researchers studying the computational properties of brain models or robotic controllers built upon large-scale spiking neural networks. It is an extension of the proof of concept for large-scale NEST simulations on the NRP successfully developed in the framework of the RoboBrain Voucher.