Title: Version 3.2.0

*Science Use case & Workflow Description and Specification*

05-Nov-21 Author Names,

|  |  |
| --- | --- |
| Partners |  |
| Institutions |  |
| Principal Investigators |  |

|  |  |
| --- | --- |
| Date | Version / Change |
|  |  |
|  |  |
|  |  |
|  |  |

This document provides a tool for scientist and developers to guide the collaborative process of transforming a free form description of a science use case into technical specifications. This is achieved by following the instructions given in the Appendix. Here, in addition to a detailed description of what each section should contain, also the concrete purpose and motivation of this document is explained.

Inhalt

[1 The Use case 2](#_Toc86846106)

[1.1 Use case description 2](#_Toc86846107)

[1.2 Diagrams 2](#_Toc86846108)

[1.3 Components characterization 2](#_Toc86846109)

[1.4 Fenix infrastructure requirements 2](#_Toc86846110)

[1.5 Additional diagrams 2](#_Toc86846111)

[1.6 References 2](#_Toc86846112)

[2 Appendix 2](#_Toc86846113)

[2.1 What to find in the use case description? 3](#_Toc86846114)

[2.2 Setting up the annotated use case diagram 4](#_Toc86846115)

[2.3 Characterization of components 5](#_Toc86846116)

[2.4 FENIX Infrastructure requirements 8](#_Toc86846117)

[2.5 Additional diagrams to be completed when needed 9](#_Toc86846118)

[2.5.1 Visualizing timescales in computational sciences 9](#_Toc86846119)

[2.5.2 Publication time scale 10](#_Toc86846120)

[2.5.3 The research time scale 11](#_Toc86846121)

[2.5.4 The experiment time scale 11](#_Toc86846122)

[2.5.5 Interactive session time scale 11](#_Toc86846123)

[2.6 References 11](#_Toc86846124)

# The Use case

## Use case description

Please fill in your use case description according to[4](#_Toc69475874) 2.1

## Diagrams

Please create your diagram according to 2.2

## Components characterization

Please characterize your nodes according to 2.3

## Fenix infrastructure requirements

Please specify your infrastructure requirements according to 2.4

## Additional diagrams

Please specify your infrastructure requirements according to 2.5

## References

Please add references used in the use case template description here.

# Appendix

This use case description and specification document provides a tool for developers and scientist to guide the collaborative process of transforming a free form description of a science use case into technical specifications. Specifications that guide the implementation of workflows fulfilling the science use case. This document should help the science case in a number of ways: its structured methodology will help to find the essential parts, and it will assist in separation of the ***m*ust-haves** and ***nice-to-haves***[1]. The specifications should result in a standalone document that can be given to new partners of the project as introduction into the science and technical details of the project. On a more abstract level, this document could be considered a contract formalizing of the expectations of both engineers and the scientists.

In the process of creating a science use case analysis document, it is important to focus on the distinction between user requirements and entailed technical implementations. A user is ultimately only interested in the functionality of a software/hardware product and not in the underlying technical details of the implementation. Separating these concerns is a non-trivial matter: This document will therefore typically be written in an iterative manner, with the document bouncing from scientist to developer getting more detailed on each iteration. It will also be a living document: Details of the project can and will change over time; Components might be hard to implement and trade-offs might be made depending on the availability of resources and technical limitations. The amount of work needed for the creation of the document might appear large, but it pays off, not only through the general benefits mentioned above but also through more direct implications, such as the early identification of potential problems and possible solutions.

Ideally, the different elements/chapters in the template should follow the here proposed order and content. This will allow comparison of different use cases and the identification of shared/overlapping functionality. This document and the accompanying PowerPoint presentation introduce a set of visual components that can be used to describe use cases and associated systems55. The symbols should cover the majority of systems encountered, but if needed, new element can be introduced. Introducing new elements should be done with caution, because it complicates the comparison between cases. The main advantage of this unified visualization scheme is the possibility of reusing software components, which saves time and work on the long run. Although the introductory chapters can be removed, it will limit the use of the documents as an introduction for new project partners.

In the next sections, the goals of the individual parts of the template will be introduced. The first section ([2.1](#_What_to_find)) details the use case description. It should provide the scientific reasoning behind the case. Section [2.2](#_Setting_up_the) explains the set of visual components that can be used to create the model diagrams. In section [2.3](#_Characterization_of_components), we provide a description table that enables the characterization of different data components in more technical details.

Section [2.4](#_FENIX_Infrastructure_requirements)  contains a list of questions specific to potential infrastructure requirements. Here, high-level needs and services can be crosschecked with the node characterizations. Section [2.5](#_Additional_diagrams_to)  provides a growing set of potential additional diagrams to be created in the process of formalizing your case. These diagrams will only be created depending on the actual need for this level of formalization.

The main document ([chapter 1](#_The_Use_case)) will contain the filled out template; for now, it contains just the titles. Other components can be copied from the appendix [chapter 2](#_Appendix). If you are describing multiple systems, resulting into multiple fully disjoint diagrams it is best to copy the template multiple times or to use different documents as this signals that you are describing multiple science use cases or workflows. This isolation will improve coherence in the descriptions and again will improve the identification of shared functionality.

## What to find in the use case description?

The workflow description is a high-level description of the use case. It is typically written by the scientists providing the main scientific content and, in line with this, the reasons why to build or use a software/hardware system. Topics that might be encountered in this section are for example: New (or better, bigger, faster) science is possible with this software and, problems and challenges encountered in the current software. Typically, the workflow is broken down in steps with partial goals.

It is advisable to keep implementation and technical details out of this section.

An example of such an ***implementation detail*** would be: “The software must be fast, to allow fast turnover of experiments. Therefore, **w*e have to use GPUs***”. Complete separation of concerns is hard to arrive at. It is one of the more complicated exercises in system design. Having a starting point is more important than being completely correct. This is one of examples in which the dialog with technical experts will help to arrive at a correct description.

An example of a **science (and not technology) centric description:** “As a researcher I want to be able to perform a large scale computational experiment. This experiment cannot be performed on my local cluster due to the size of the parameter space I want to explore. The analysis of the results needs to be performed in my local institute due to A and B. The access of the results should be structured based on X and Y. “

Two widely different technical solutions would support this case. (These would then be added to the later chapters of this document):

1. Analysis of results on a virtual machine with data staying at a central location. Results are selectable via a database, accessed via a web interface.
2. Transport of results to the local cluster with processing on the local machines with the data stored in clearly labeled directories.

Which of these solutions is implemented can now be selected based on available resources, software limitations, etc.

## Setting up the annotated use case diagram

An annotated use case diagram is a relatively freeform graphical depiction of the textual description as detailed in this section. We would suggest using the diagram components as shown in Figure 1 (next page). This will allow easy comparison between different use case descriptions. The flowcharts in this document follow the practices as described in [2], [3].

To prevent cluttering of complicated workflow we suggest the following:

* Make use of specialized symbols to allow visual distinguishing of salient features (GUI would be an example).
* Use only a small pictogram for data objects annotated with a number.
* Use the suggested locations for the connectors: Control at the top. Inputs from the left or bottom. Outputs leave on the right side.

Reiterating, these are suggestions, the diagrams are in principle freeform, and not all symbols might be used in your specific use case. An editable version of the diagram will accompany the current document as a PowerPoint presentation. It also includes example diagrams. Figure 2 shows an example diagram. Not all components are yet marked with a number in this diagram illustrating that a work in progress diagram can already be insightful.



Figure : Overview of suggested symbols for a use case diagram. The symbols are based on [2], [3]. The symbol for Graphical User Interface (GUI) is a combination of processing station and data object. A suggested typical data and information flow is shown. Additionally, a simple bandwidth range is depicted.



Figure : Example diagram of an abstract machine learning workflow with user interaction. Not all components are yet marked with number illustrating that a work in progress diagram can already be insightful.

## Characterization of components

This section targets a characterization of each component that is depicted in the annotated use case diagram. This is achieved based on two different types of tables.

### Overview

The first table serves as an overview table, which is particularly important for very complex workflows. Each row corresponds to a component of the diagram. Additionally, the following information can be entered be provided:

* **Number of the component: The number given to this component in the annotated use case diagram**
* **Name of the component: Short, meaningful name of the component**
* **EBRAINS component**: Is this component based on tools/services provided by EBRAINS and/or will it be integrated into EBRAINS (e.g. a model is planned to be published in the Knowledge Graph of EBRAINS) A ‘yes’ or ‘no’ answer is enough here.

**Potential challenge: Do you foresee any challenges for this component?**

* **Maturity Level:** A rough rating to describe how far the component is already progressed. This ideally improves in the course of the project, optimally ending at number 10.

**Maturity Levels**

1. Projected component without requirements know
2. Requirements collected
3. Concrete implementation plan
4. First implementation (not connected to workflow)
5. Tested first implementation
6. First implementation as part of a workflow
7. Tested version in a workflow
8. Documentation ready
9. External review (HLST etc.)
10. Integration in EBRAINS, if applies, following TC recommendations.
11. publication

We suggest the following layout:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No | Name | EBRAINS? | Potential Challenges  | ML |
| 1  |  |  |  |  |
| 2 |  |  |  |  |

This second set of tables serves for components that need a more detailed description. Here, every single table corresponds to a component format with typical information points listed. The entries are typically split into different sets:

* The ***base*** information, which gives a general description regarding the component. The description is typically at a user/functional level.
* **Technical specifications** that are known at this stage. The use case is not yet solved. Thus, this information will be added incrementally and optionally validated at some stage by domain specialists.
* **Current solution (optionally)** that one is aware of. Not all information might be available, so we recommend filling in what is known at this stage. Having a starting point for a dialog is more important than having perfect information, especially for the beginning stages.

### Data objects

|  |
| --- |
| Data object: ***number in diagram***, name |
| ***Base Information*** | General description of what data is stored. Potential additional information: formats, metadata, database requirements |
| ***Technical specifications*** | Please select:* Transient (temporary): data discarded on simulation completion or when later processing steps are concluded
* Short-term (campaign): data used throughout the execution of the scientific workflow
* Permanent (forever): data outliving the machine used to generate it
 |
| ***Current solution(optional)*** | Name |
| URL to additional information |
| Limitations |

### Data transport

|  |
| --- |
| Data transport: ***number in diagram,*** Name |
| ***Base information*** | General description of what data is transported.Potential additional information: data access patterns (request rate, transfer sizes) |
| ***Technical specifications*** | (est.) Maximum required bandwidth |
| (est.) Average required bandwidth |
| Interface requirements |
| Additional information |
| ***Current solution (optional)*** | Name |
| URL to additional information |
| Limitation |

### Data ingest/GUI

|  |
| --- |
| Data ingest: ***number in diagram***, Name |
| ***Base information*** | Description of input data sourcePotential addition information: description of data introduction, e.g. upload? scanner characteristics? simulation characteristics? |
| ***Technical specifications*** | Characteristics of data: formats, loads, bandwidths, latencies, transports |
| Additional information |
| ***Current solution******(optional)*** | Name |
| URL to additional information |
| Limitation |

### Data repository

|  |
| --- |
| Data Repository: ***number in diagram***, Name |
| ***Base Information*** | Classification of the data objects (see below) |
| Access control requirements |
| Access requirements |
| Data availability requirements |
| ***Technical specifications*** | Maximum and average capacity requirements |
| In case of repositories for permanent data objects, i.e. repositories where data is accumulated, provide maximum capacity requirement as function over time(est.) In terms of size and file number. |
| Additional information |
| ***Current solution*** | Name |
| URL to additional information |
| Limitation |

### Processing stations

|  |
| --- |
| Processing station: Name |
| ***Base Information*** | General description of data processing |
| ***Technical specifications*** | Hardware architecture requirements(est.) computational needs |
| Additional information |
| ***Current solution******(optional)*** | Name |
| URL to additional information |
| Limitation |

## FENIX Infrastructure requirements

This section of the template will map from the FENIX HPC infrastructure to the science use case. For each infrastructure service, we ask specific questions about how this service might be used for your science use case. There will be an overlap of information provided through the annotated use case diagrams. This duplication is **intended.** It will allow consistency checks. This avoids the need of fixing the mapping between the case and specific infrastructure services at a later stage.

The following definitions and additional information are available at:

<https://fenix-ri.eu/infrastructure/services>).

[*Interactive Computing Services*](https://fenix-ri.eu/infrastructure/services/interactive-computing-services)

Quick access to single compute servers to analyze and visualize data interactively, or to connect to running simulations, which are using scalable computing services.

[Scalable Computing Services](https://fenix-ri.eu/infrastructure/services/scalable-computing-services)

Massively parallel HPC systems that are suitable for parallel brain simulations or for high-throughput data analysis tasks.

[Virtual Machine Services](https://fenix-ri.eu/infrastructure/services/virtual-machine-services)

Service for deploying virtual machines (VMs) in a stable and controlled environment that is, for example, suitable for deploying platform services like the HBP Collaboratory, image services, or neuromorphic computing front-end services.

[Active Data Repositories](https://fenix-ri.eu/infrastructure/services/active-data-repositories)

Site-local data repositories close to computational and/or visualization resources that are used for storing temporary replicas of data sets. In the near future, they will typically be realized using parallel file systems.

[Archival Data Repositories](https://fenix-ri.eu/infrastructure/services/archival-data-repositories)

Federated data storage, optimized for capacity, reliability, and availability that is used for long-term storage of large data sets, which cannot be easily regenerated. These data stores allow the sharing of data with other researchers inside and outside of HBP.

|  |  |
| --- | --- |
| **Infrastructure service** | **Questions to address** |
| Interactive Computing Services | * Which parts of the workflow require such services?
* What is the expected typical duration of interactive sessions?
* What software stacks need to be available?
* Is it possible to define memory capacity requirements?
 |
| (Elastic) Scalable Computing Services | * Which parts of the workflow require such services?
* How/Is horizontal scaling implemented in your workflow?
* Do you need high availability services?
* Do you need replication of your workflow on multiple sites
 |
| Virtual Machine Services | * Which parts of the workflow require such services?
* What is the longest duration process performed in your virtual machine?
* Do you have monitoring capabilities in your service?
 |
| Active Data Repositories | * Which parts of the workflow require such services?
* Do you need this data from multiple places?
* Should your data be backed-up automatically?
* Do you need this data to be replicated at multiple sites?
 |
| Archival Data Repositories | * Which parts of the workflow require such services?
* How long should your data be accessible?
* Do you need your data to be replicated at multiple sites?
 |

## Additional diagrams to be completed when needed

### Visualizing timescales in computational sciences

EBRAINS has amongst its goals the creation of an e-science infrastructure to support computational neuroscience in the broadest sense. Due to the inclusive, large and broad scope of this endeavor and the multi-scale nature of the human brain the science, methods and computation solutions are equally varied:

* From measuring the current on a synapse, to tracking patient performance on mental tasks over time.
* From simulating ionic transport in a membrane junction, to exploring the neuroscientific foundation of lifelong learning.
* From deploying an analysis script on a supercomputer, to laboriously writing a paper while collaborating with your colleagues.

The juxtaposition of these activities shows the scale of the challenges the HBP has embarked on. In this section, we acknowledge the importance of the science time scales involved in the HBP. Different time scales have different challenges and call for different solutions. Figure 3 (next page) introduces the stacked time scale diagram, which helps to formalize the different timescales in your science use case. The accompanying PowerPoint presentation contains an example of an instantiated diagram.

 

Figure : Stacked timescales. The top time scale describes the longest timescale, here measured in years. At this this scale we expect to see the overarching project outline, and related project management actions. At this level, the long term storage of the results for reproducibility (in some instances up to 10 years) is found. On shorter time scale (measured here in months), we find the research time scale. At this level, actions like exploration of available HBP Atlases might be placed, selection of data, but also the storage of results. On smaller scale, looking at individual experiments more technical action and moments occur: setting of parameters, scheduling jobs. At the smallest time scale, we zoom in to the details of an interactive HPC session where we want to stage our jobs and interact with the HPC systems.

This breakdown of the complete life cycle of science performed in the EBRAINS research infrastructure that this document aims to support, gives equal weight to challenges and solutions at the different timescales. It relates the science phases at a widely different conceptual scale. Additionally, it allows the ordering of actions and gives a natural grouping of actions and concepts.

The diagram as presented in Figure 3 breaks down a complete end-to-end workflow in four different time scales. It is important to note that an instantiation for a specific use case might have a different number of scales or different units on its axis, or might not have information available on a specific time scale.

### Publication time scale

The largest time scale spans multiple years. Items to be found at this level could be the creation of user groups, sharing of research results between users, and the access of provenance data for reproduction of data. The current version of the Collaboratory performs a number of the functions of this time scale. An outside scientist might typically first encounter functionality as provided by EBRAINS. The information at this level is typically expected to be public at some stage. At this level, educational material and public relations material have their home. The functionality of this layer is best compared to existing science portals, or even publication and data storage sites. The technological solutions could be at the levels of websites, wikis, GIT repositories, and databases. Most activities in the EBRAINS would share this level and thus multiple solutions are already provided.

### The research time scale

At the timescale of months, we find the ‘research time scale’. Functionality at this level would include high-level tasks like user management or supporting the need for private data sharing inside research groups.

This is a collaborative time scale with data typically accessible for multiple users: access to data as produced in different experiments, exploration of the results and reading of data for publication. The current version of the Collaboratory performs a number of the functions at this level, e.g. data and content sharing via wikis and GIT repositories.

Although the actions and requirements on this level are becoming more specialized (e.g. requirements of Atlas exploration are completely different from the needs of selecting algorithms for steering a robot), also a number of technical solutions are expected to be shared, for example authentication, data access, and data transport.

### The experiment time scale

At the time scale of days, you find the experiment time scale. Access to the functionality is typically via components/functionality at higher time scale. Access to this functionality should be expected to be username and password protected. In contrast to the research time scale, activities are typically for a single user, with the exception of accessing shared data resources. The functionality at this level will be task-specific: setting of application-specific settings, creation of HPC jobs and interactions with GUI’s like the Neurorobotics Platform (NRP). It is at this level that a researcher is expected to spend the majority of this time. Because a majority of the functionality of the HBP will be located at this level, anonymous access for potential users could be supported if needed.

The actions at this level are not necessarily application-specific: e.g., Jupyter Notebook editing, analytical methods like machine learning or simple Numpy operations. It is at this level that good computational workflow support is essential. Typically expected is the creation of HPC job scripts and data transport to and from central storage locations.

Technical solutions for these tasks are currently performed by the Collaboratory. Further workflow management support and integration offered via UNICORE or SNAKEMAKE will greatly increase the usability at this level. The usage of existing solutions allows building on existing functionality like provenance tracking and HPC/workflow monitoring.

### Interactive session time scale

EBRAINS and its supporting infrastructure support interactive access to real-time resources: HPC simulation, neuromorphic hardware, the NRP and interactive exploration of large-scale datasets in the HPC brain atlases. The NRP has an existing toolchain allowing interactive control of robotic experiments. The Brain atlas provides interactive access to petabytes of integrated data. More and more simulators are in the integration process and will allow scientists to adapt model parameters on the fly in the future. Al these efforts are unique in the world but also provide unique challenges: Monitoring and provenance tracking of these multi components workflows, in which not a single component should fail, is an unresolved challenge.

## References

[1] *MoSCoW Analysis (6.1.5.2)*. International Institute of Business Analysis, 2009.

[2] “Flowchart Symbols Meaning | Standard Flowchart symbol images and usage.” [Online]. Available: https://creately.com/diagram-type/objects/flowcharts. [Accessed: 17-Aug-2017].

[3] “Flowchart Symbols and Notation | Lucidchart.” [Online]. Available: https://www.lucidchart.com/pages/flowchart-symbols-meaning-explained. [Accessed: 17-Aug-2017].

[4] “UseCaseDescription\_and\_Specification\_v1.”