IMI2 821520 – ConcePTION

ConcePTION

WP7 Information and data governance, ethics, technology, data catalogue and quality support

D7.11 Wiki page with description of tools and GITHUB repository

Lead contributor	Daniel Thayer, Swansea University (23 – USWAN)
	D.S.Thayer@swansea.ac.uk
Other contributors	Krishna Bodapati Menon, Novartis (38 – NVS) Rosa Gini, Agenzia Regionale di Sanità (10 – ARS) della Toscana

Document History

Version	Date	Description
V0.1	11 Nov 2020	Contents
V0.2	26 Mar 2021	Initial Draft
V0.3	26 Mar 2021	comments, Rosa Gini
V1.0	28 Apr 2021	Final version following review by WP7 members and ConcePTION Managing Board
V2.0	31 July 2021	Final Version updated following IMI feedback on template format

Abstract

The R developers' group within Work Package 7 for the ConcePTION project have created a set of solutions for storing code and associated documentation, in order to support development of analytic tools for the project, release to users, and user support. The goals of the deliverable were:

- Support the development of new analytical code within ConcePTION.
- Be able to involve developers both within ConcePTION and from outside organizations.
- Divide development into logical subprojects to simplify the process.
- Be able to include code developed elsewhere and shared with ConcePTION, including ongoing projects owned with other groups.
- Facilitate code sharing and reuse within ConcePTION and beyond.
- Be able to reuse and repurpose individual analytic components.
- An easy installation experience for users within ConcePTION.

To these ends, we have created a Github organization with a number of repositories for individual tools, and a set of working practices to use this resource effectively to meet the needs of the project. This resource can be found at https://github.com/IMI-ConcePTION

Github is a popular cloud-hosted service for Git, the industry leading version control system for software development. The benefits of Git to this project include tools facilitating collaboration, as well as storing a complete history of the code throughout its development. To these core features, Github adds a web presence that facilitates sharing code, as well as issue reporting and management.

Repository architecture for functions

Functions are reusable, general tools that implement functionality and serve as the building blocks of analysis scripts, etc. Functions are implemented as R packages. The division of functions into packages is based on logical relation of the functionality and the development team involved; each R package may have one or more functions.

Each R package is hosted in its own Git repository within the ConceEPTION Github organization. The repository has its own development team with appropriate permissions, and can include developers who are members of ConcePTION or external developers where appropriate.

It may be useful to share other associated information along with an R package, such as usage examples, or example (synthetic) data. Such information can be included within the Git repository as well, making it accessible to users.

Relationships between functions can be defined as R package dependencies; R will automatically download dependency packages of a given package, including from Github where the repository location is defined—enabling a simpler experience for the end user.

Several R packages have been created already or are in development, including CreateFlowChart (https://github.com/IMI-ConcePTION/CreateFlowChart) and CountPersonTime (https://github.com/IMI-ConcePTION/CountPersonTime).

For development and reusability purposes, it is ideal to create a number of R packages that encapsulate discrete sets of functionality. However, we also had the requirement of making installation of the required packages as easy as possible for users within the ConcePTION project. Therefore, we created the concept of a light wrapper package: this package, called ConceptionTools (https://github.com/IMI-ConcePTION/ConceptionTools), contains all functions within the ConcePTION project as dependencies, enabling the user to install them all with a single command. The ConceptionTools package includes both packages developed within ConcePTION and packages developed by outside organizations that are

used as part of the development process for ConcePTION, for instance CreateConceptSetDatasets (https://github.com/ARS-toscana/CreateConceptSetDatasets)

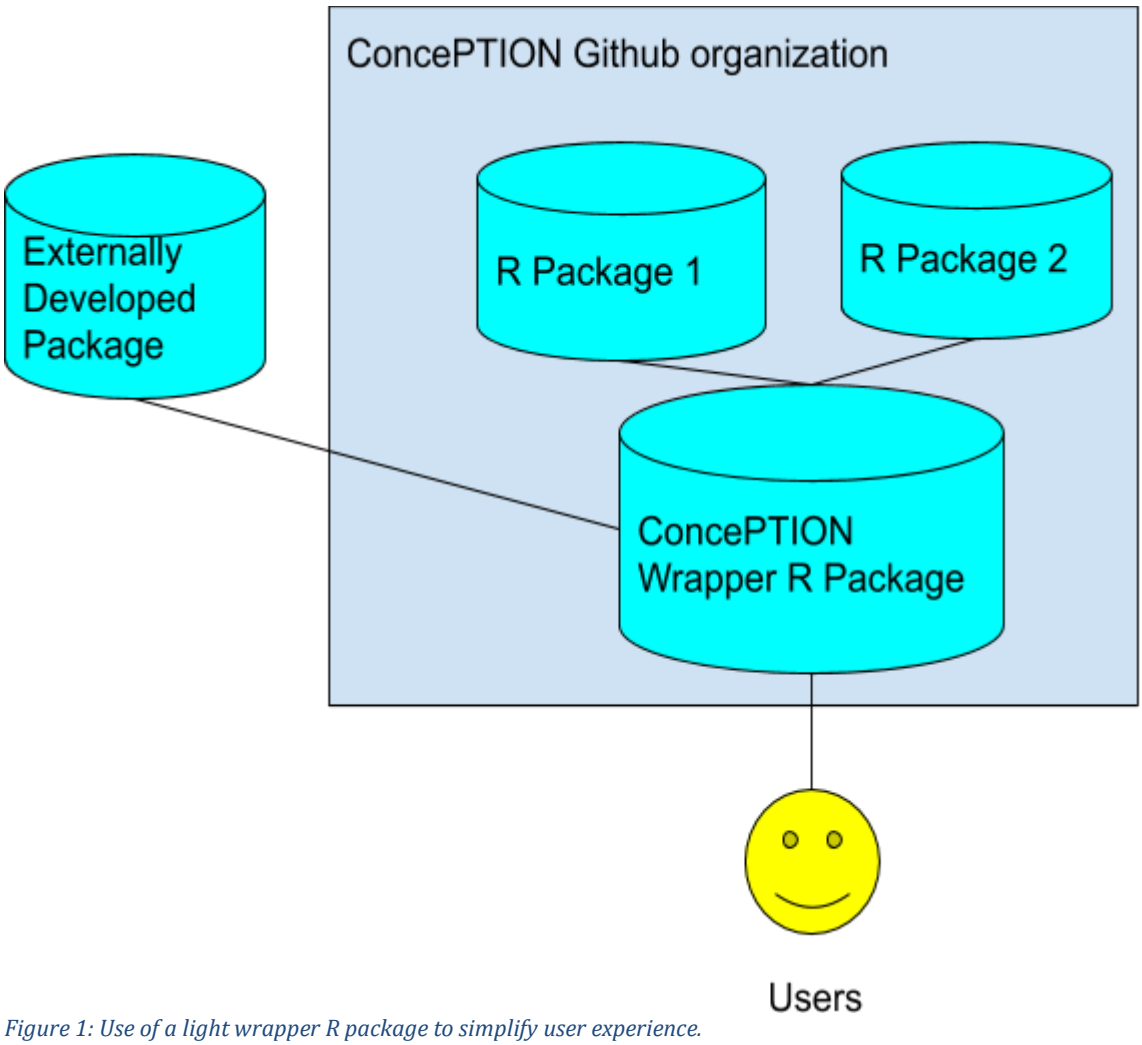

Figure 1: Use of a light wrapper R package to simplify user experience.

Repository architecture for scripts

In contrast to functions, scripts are bespoke pieces of code created for a single purpose (such as the analysis for a particular study). Scripts may rely on functions, which can be included through the standard R mechanisms for including packages.

Scripts, as with packages, are each stored in their own Github repository within the ConcePTION organization. Examples of scripts that have already been created and stored in Github include a 'sample' script (https://github.com/IMI-ConcePTION/sample-script) aimed at representing a typical script in the ConcePTION pipeline, as introduced in the Deliverable 7.5.

In the repository, along with the code, a wiki is populated, with the following structure

1. protocol: link to the protocol and/or SAP of the study
2. data models of the datasets: data model of all the final outputs of the script, as well as of the intermediate datasets generated by the steps of the script
3. Structure of the script: the script complies with the format introduced in Deliverable 7.5, including a main script, parameters, functions, and steps. In this section, the specific structure of

this script is depicted graphically, see Figure 2. in the graphical representation, steps and datasets link to the corresponding sections of the wiki.

4. Actions in each step: each step of the script is described in terms of input, output, and action, with link to the corresponding code in the code section of the Repository.
5. Parameters: description of the parameters involved in the script
6. Instructions: how to download and execute the script.

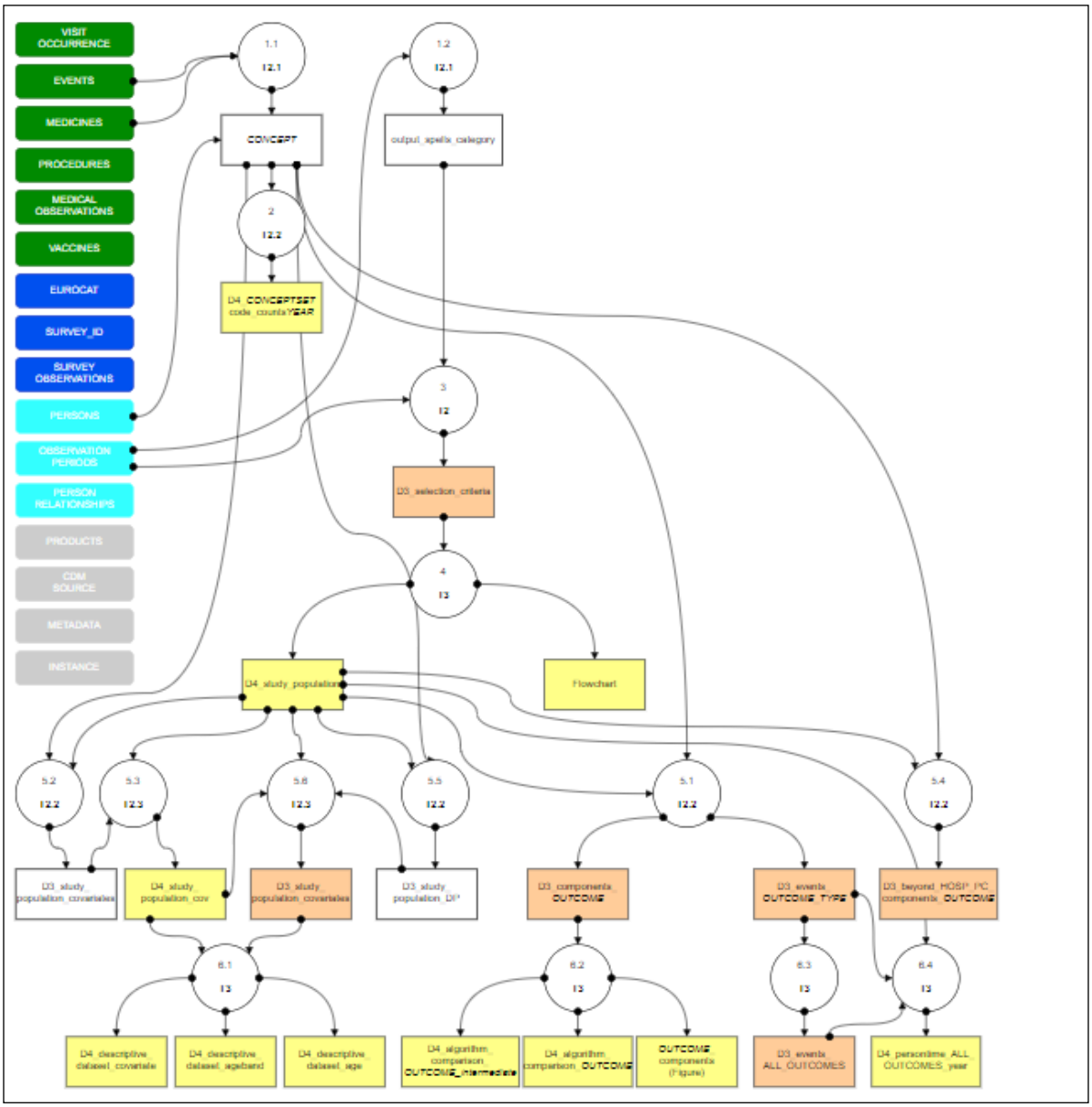

Figure 2. Graphical representation of the structure of a script of the ConcePTION pipeline, as represented in its GitHub repository. Steps and datasets link to the corresponding sections of the wiki.

Development Workflow

Development within Git is conducted using a simple branching workflow. In Git, multiple branches, or versions of the code, can exist simultaneously. The main branch contains the released version of the code that users would access and use. Development to add new features to the code would occur in one or more separate development branches. This allows developers to work independently on a new feature without disrupting the availability of the released version that users need. Once development has progressed enough that a new version can be made available, a development branch can be merged

back into the main branch. An experienced team member would review the updated code to ensure that it is ready for release and approve the merge.

Besides being convenient for development, this workflow also allows quickly resolving problems if a bug is discovered in the production code. A quick bug fix can be applied directly to the main release branch without waiting for the longer development process to complete.

Issues

Github's issue management system can be used for multiple purposes. First, to support the development workflow: investigators and/or developers can create issues describing new requirements to be implemented, or developers or users can create issues when bugs are discovered. Second, at the script level, development can be organised in issues since its inception, for instance once a step is specified its development can be assigned as an issue to a specific developer or group of developers, pointing initially to the wiki page, and discussions around the development of that step can be developed and possibly include the investigators in the process. Similarly, development of data models of input/output datasets can be discussed in this forum.

New updates to the code can be tied to the corresponding issues and marked as resolving or completing them as appropriate.

Documentation

The combination of two resources is used to document ConcePTION project code. The built in R package documentation functionality is used to document functions released as packages. This integrates into the R programming environment, so that users can easily access help information while they are developing code using these packages. Additionally, the GitHub wiki feature serves as a location for high-level, web-facing documentation about project code (see for instance the 'script' wiki described above).

Conclusion

We have developed methods for creating and sharing code to support the needs of the ConcePTION project, both to successfully manage a development process that includes many participantes, as well as to meet the needs of the code's users (such as data providers). The combination of R package functionality with Github, along with associated documentation features, will enable us to deliver and support the software tools needed for our research.