

Drug Disease Model Resources
ddmore



DDMoRe Workflow "Thoughtflow"

Project Plan Version 1.2

Created by Mike Beer (scinteco) based on input from DDMoRe

Contents

Management Summary	3
Objectives.....	4
Work Breakdown Structure	6
Thoughtflow work packages (TFWPx).....	6
Phases of the Project.....	7
.....	7
Phase 1 Functions	7
BWF	7
Import into repository.....	7
Export API	7
Phase 2 Functions	7
Phase 3 functions	8
History.....	9
Rationale	10
User Expectations	10
Project Structure & Timeline	11
Functional Overview.....	11
Project Components	11
Timeline	13
Workflow Visualization – User Interface	14
Login.....	14
After login: news	15
Projects	16
Project visualization	17
Step visualization.....	18
Re-Run.....	19
Importing a file	20

Cloning a project.....	21
Edit a script.....	22
Suggested Team Roles	23
Contingency Considerations.....	24
Risk Factors.....	24
Potential actions:.....	25
Required Proposal Items from scinteco.....	26
TFWP7 BWF – Phase 1	26
TFWP8 BWF – Phase 2	26
TFWP9 BWF – Phase 3	26
TFWP10C	27
TFWP11C	27
TFWP12C	27
TFWP13C	27
Open Issues	28
Appendix	29
Mail from Gareth Smith (April 28 th 2016).....	29
Mail from Andrew Dunn (April 21 st 2016).....	30
Appendix: DDMoRe Project Work Packages	31

Management Summary

In February 2016 scinteco was approached by the DDMoRe¹ consortium, to support in creating a cross-platform tool which will facilitate robust tracking, reporting and replication of elements of a PKPD project, either in part or as an entire analysis. It will be nominally independent of the other DDMoRe tools, but will be able to benefit from them if they exist. A first prototype is expected for August 2016.

The goal of the Workflow/Thoughtflow workstream is to further develop the tool for tracking and reporting the provenance of activities, entities and decisions making up a pharmacometric data analysis project, with a view to facilitating updates of aspects of the analysis, or the entire analysis, after changes to inputs, and to facilitating reporting: audit logs, QC checklists and run records are three central examples.

Main features for a minimum deliverable –

- Track entities, activities, decisions within a project, and how they relate to one another
- Basic visualizations of these relationships
- Extraction of information for reporting

References to the part that scinteco will provide is specified in Project Components as "BWF" (Browser Work Flow). The whole DDMoRe-workflow package is referenced as "thoughtflow".

¹ Current DDMoRe participants: Pfizer, Lilly, Merck, AstraZeneca, GSK, Mango, Leiden University, Cyprotex

Objectives

Thoughtflow is supposed to support the following key values and features:

Value	How DDMoRe delivers this
Traceability	Workflow tool records all individual steps throughout analysis with time stamps and dependencies
Clarity	On-the-fly and post-hoc visualisation of the steps of the analysis and their interdependencies
Audit trail	Workflow tool may be used to generate a complete audit trail for an analysis
Decision tracking	Decisions may be documented at pre-defined stages of model development; visualization of decision tree
Facilitated review process	Inspection points may be defined for post-hoc quality review from initial data set to final report
One-click re-execution and reproducibility	Upon changes made to key entities (such as datasets), all dependencies may be re-generated using a single click; entire analysis may be repeated in this way
"Lab Book"-like documentation	Workflow tool automatically collects information that can be queried in a reporting, scientific, QC or audit context
Stand-alone installation	Runs independently of other DDMoRe software components, although delivers benefits if these are present

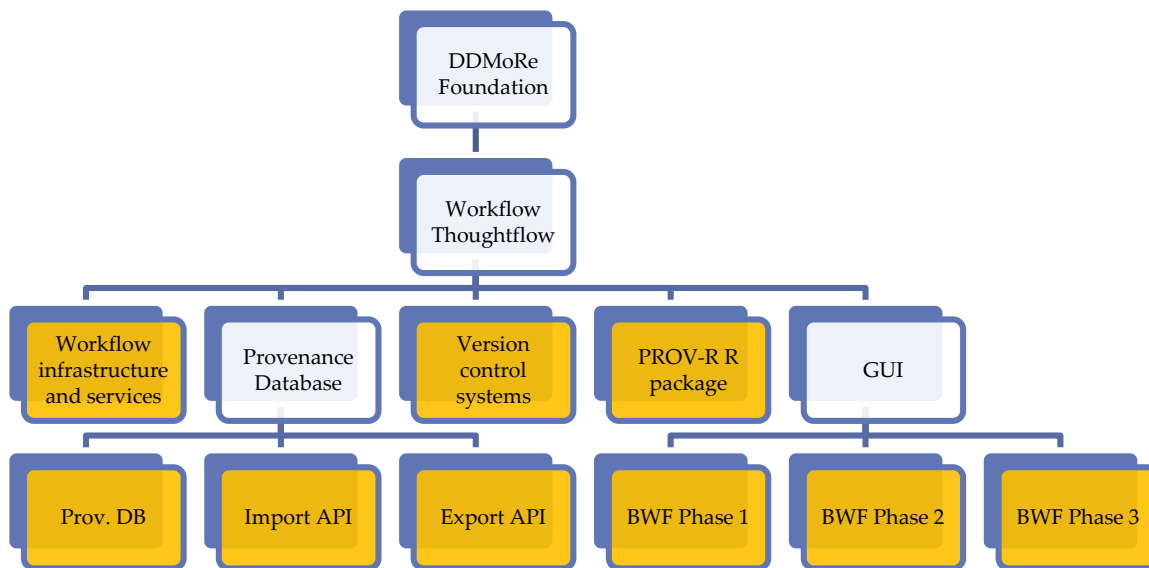
Feature	Description
Tracks provenance	Recording of relationships between components of an analysis (scripts, models, outputs, decisions, ...), minimizing burden on the user
Artifact version control	Detects when a component has changed, records the nature of the change, and determines its impact on other components
Updates analysis components to reflect upstream changes ("re-running")	When changes to an analysis component (such as a data file) have been made, all dependencies may be updated at the click of a mouse
Exports run records and audit trails	Run records and audit trails may be assembled and exported in a convenient format for reporting, minimizing analyst need to do this manually
Assembles report components	Key graphs and tables can be prepared and assembled in a convenient location and format for reporting and QC
GUI facilitates project and provenance visualization	A streamlined, portable GUI centralizes all functionality without getting in the way
Independent but synergistic	DDMoRe tools are not required for core functionality, but their availability will bring benefits
Built on standards	Underlying technology is built on widely-adopted standards for defining provenance (PROV-O) and programming tools
A solid foundation	The tool will be designed to support further development and expansion

BWF will need to handle the following data:

- Files:
 - Input & output files (e.g. CSV, output.txt, images)
- Meta Data
 - Run records
 - Audit trails

Work Breakdown Structure

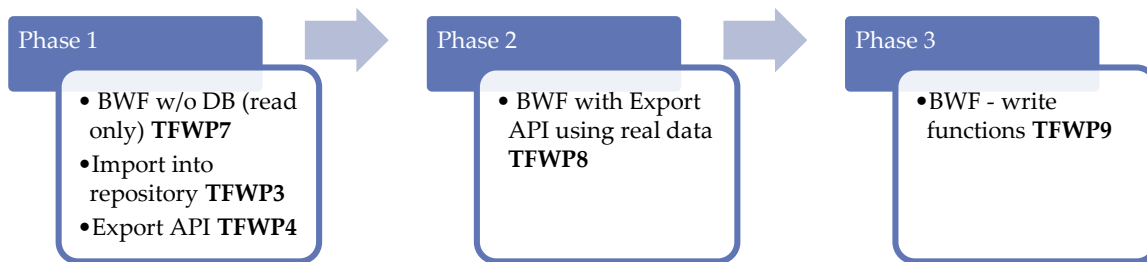
The project consists of the following work packages:



Thoughtflow work packages (TFWPx)

- 1 **TFWP1: Workflow infrastructure and services:** provides vital backend services (e.g. communications, "watcher" component) – prereq for BWF Phase 2
- 2 **Provenance database:** where provenance information is stored – prereq for BWF Phase 2
 - a. **TFWP2:** Provenance Database (Fuseki)
 - b. **TFWP3:** Import API into Fuseki and GIT
 - c. **TFWP4:** Export API
- 3 **TFWP5: Version control systems:** control of entity versioning – GIT, existing
- 4 **TFWP6: PROV-R R package:** provides functions for integration with R, and other DDMoRe tools and processes
- 5 **GUI:** frontend to facilitate user interaction with the system
 - a. **TFWP7:** BWF Phase 1
 - b. **TFWP8:** BWF Phase 2
 - c. **TFWP9:** BWF Phase 3

Phases of the Project



Phase 1 Functions

In the first phase the following 3 work packages need to be created:

BWF

- Login
- Search in Database
- Display results
- News (who changed what)
- Dashboard view (project going on..) manager view

Import into repository

Files and their metadata can be transferred to the repository. Files move into GIT, Metadata is generated automatically and stored in Fuseki.

Export API

The export API is a REST interface that allow other applications to retrieve data from the repository. The API requires authentication and is linked to the user management.

According to the documentation available is it unclear, if there is a centralized user management (e.g. LDAP) that can be used or if this service (including administration) has to be created.

Phase 2 Functions

Phase 2 relies on having a fully functional import into the repository on one hand and on having a working export API on the other hand. Instead of using sample data (files in phase 1) now real data is fetched from the repository. User authentication is done against a central database.

Phase 3 functions

In addition to the read only functions the following features are added:

- Metadata:
 - Mark Base model
 - Mark final model
 - Add metadata:
 - passed QC
 - passed (peer) review
 - add comment
- Manipulation of scripts
 - Edit and store a script
 - Re-Run a script
- Invite people for review

History

The workflow (WF) component is planned to be an integral part of the DDMoRe tools ecosystem. Early in the project, Cyprotex proposed building a workflow component around Taverna (www.taverna.org.uk). The early iteration of this WF component, demo-ed at the autumn 2013 consortium meeting was not taken up by the consortium and was rejected by stakeholders as a viable solution.

At this point it was believed that implementation of the workflow required a lot of existing DDMoRe components to be functioning properly (e.g. converters, modelling languages, output encoding, etc). In late 2013 and early 2014 large parts of the DDMoRe system were unavailable. This made development of a working WF component impractical at that time and the majority of the early software was immature. The original designs were centred on chemoinformatics and systems pharmacology, which did not match well with the population PKPD modelling at the core of DDMoRe.

Much time was spent trying to resolve different WF viewpoints within the consortium, with varying levels of success. In Spring 2014, development work on the WF component ceased within Cyprotex. WP2 decided to stop WF development work as developer effort was required in WP5 to extend and improve converter technology.

WP2 set 4 infrastructure requirements that had to be in place before WF development would be resumed. These 4 criteria were as follows:

1. Availability of language converters.
2. Stabilisation of the modelling language specification.
3. Simplification of the model execution environment.
4. Specification of Output encoding

WF development was continued by other members of WP2, specifically Mango Solutions, who as the IDE main developer, created components for model execution centred around the Taverna WF engine and the DDMoRe standard execution environment (SEE). Outputs of this work were overseen by members of WP2 and a first release was demonstrated to stakeholders in Autumn 2014. To improve the WF design, EFPIA initiated a review process to refine the DDMoRe vision for WF and gather more concrete requirements. Most members of this review process were heavily engaged in other work packages but an expectation was received by WP2 in December 2014. Upon reception of this document, the 4 technical requirements for continuing WF work were considered to be met.

Development work for WF was re-initiated by WP2 members. Building on the previous WF work, a technical review was initiated at the close of 2014 so that the project could capitalize on the suite of advanced DDMoRe tools that had become available. During the last quarter of 2014, a WF scripting host was created, built on MDL/TEL and PharmML, which will be used to create a toolkit of elements for WF definition. Essentially, the toolkit is a graphical implementation of the Task Execution Language (TEL). This was demonstrated at the mid-term review followup in late 2014, and has been used in training, and development in ongoing.

The workflow engines that were described in the description of work (i.e. Knime [www.knime.org], Taverna, Kepler [kepler-project.org] or YAWL [www.yawlfoundation.org]) and more have been investigated extensively. After discussions with the users represented in the EFPIA working group and re-evaluating the user requirements for a workflow engine it became clear that these engines only cover

part of the functionality required by EFPIA partners in order to record, display and replay all the activities that users undertake on a day to day basis.

DDMoRe has identified that the W3 Consortium provenance ontology (PROV-O, www.w3.org/TR/prov-o) could underpin the data structure to store activities and the relationships between “data and modelling resources” (DMRs) that are developed during an analysis. The framework will be extended to capture user actions and events, and store them in this data structure. This will allow the possibility to visualize, traverse, filter, query and rerun the analysis, independently of the host system or workflow engine. This task is new but will resolve the issues around the workflow and even add flexibility to the system, for instance providing a migration strategy from existing legacy systems.

In June 2015, the WP team was updated with the addition of new EFPIA co-leads in order to refocus the WP on the needs and expectations of industry.

The tool is focused on the management of work, as opposed to workflow. While existing tools are perfectly suited to managing workflows, none is specifically designed for the management of work and of provenance.

Rationale

There are several information management tools (like Taverna, Knime, Activiti, Kepler, Pipeline Pilot, etc), but none, however, was determined to be explicitly fit for purpose, which is provenance and *work* management. Existing tools work appropriately for managing tasks, but not provenance in the context of the DDMoRe project.²

User Expectations

Users expect a fully-functional desktop tool, able to be deployed in their environment, and flexible enough to accommodate their ways of working, which may be at variance with those of others. The tool must be unobtrusive, easy to use, intuitive and - above all - stable.

² Scinteco’s improve repository would fulfill the technical requirements, but as DDMoRE needs to deliver software with open source licenses it cannot be packaged with DDMoRe.

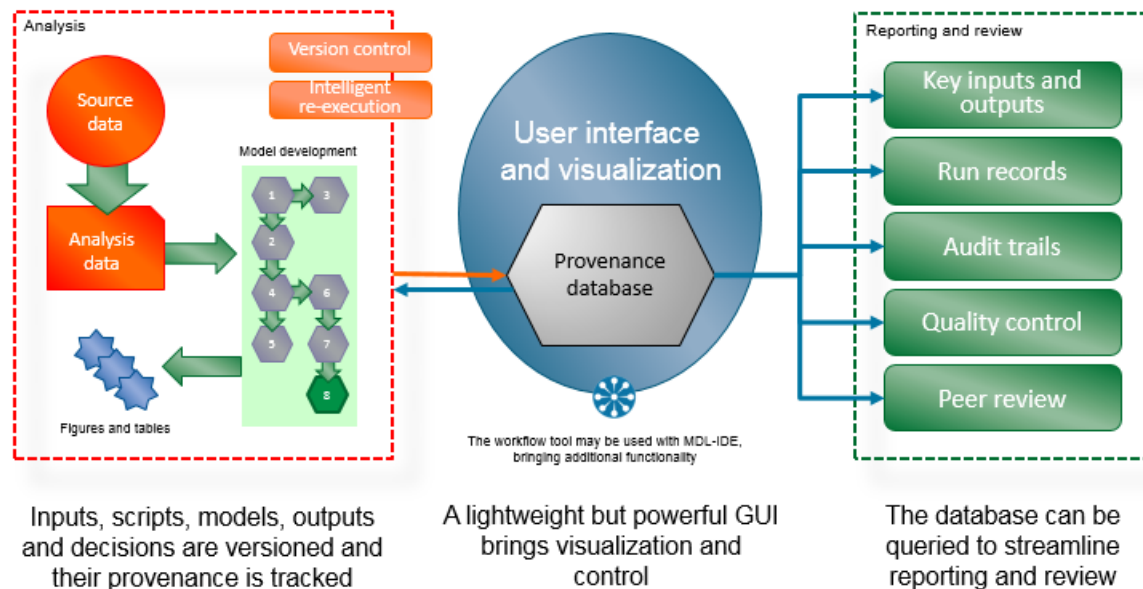
Project Structure & Timeline

Functional Overview



Pharmacometric workflow

ddmore Drug Disease Model Resources



Files are under version control by using GIT, corresponding metadata is stored in a provenance database. The provenance database is a standalone resource that can be combined with e.g. the DDMoRe eclipse-based user interface or with other tools. Several interfaces provide access to this data – for the user a browser based interface is provided. The data can be used to provide information about key inputs and outputs, run records, audit trails but also to enable QC and peer reviews. However, the first phase focuses on reading information out of the repository. For QC/review and editing of scripts and rerun write-access to the repository is needed (Phase 2).

Project Components

The workflow repository consists of 2 main components.

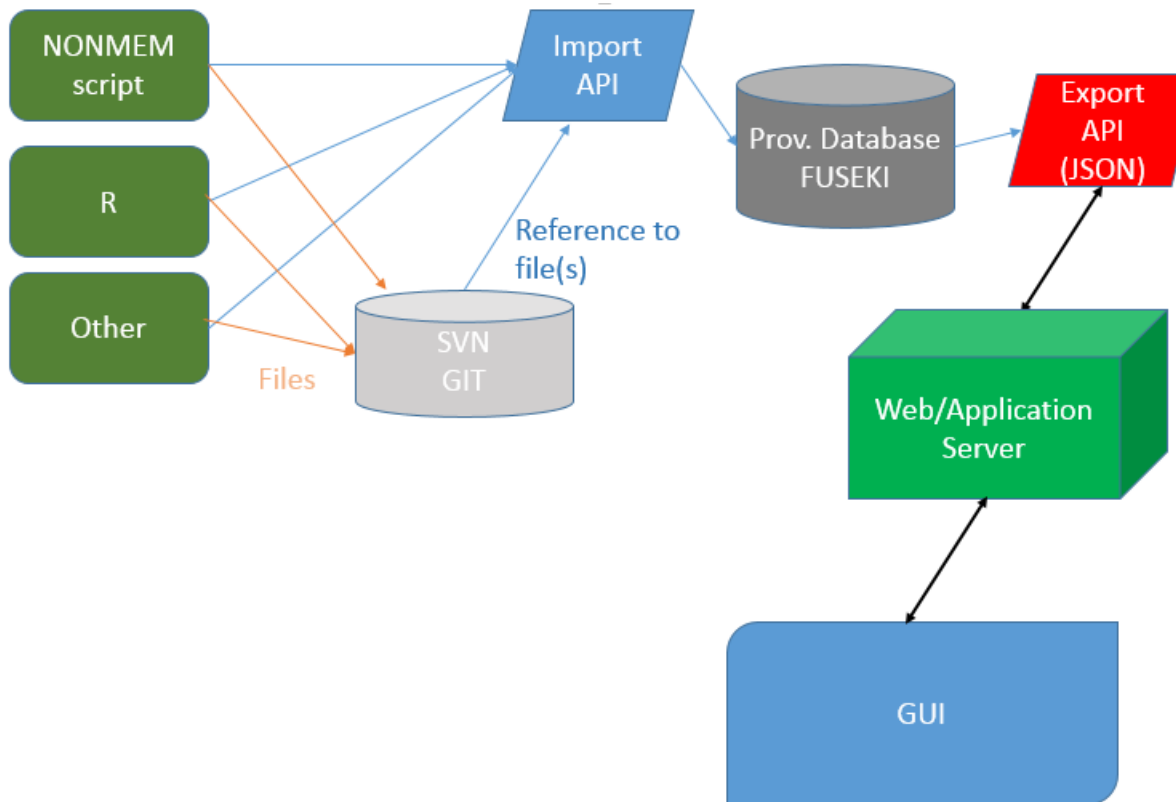
- GIT is used to store files (and their versions)
- Apache Fuseki is used to store related metadata

Data is formatted in JSON.

The project consists of 4 parts:

- 1) Creation of files out of NONMEM and R. The files are transferred to GIT. Metadata for Fuseki is generated by background jobs. The user has no direct access to the Fuseki Database.

- 2) Any tool should be able to provide data for the GIT/Fuseki repository.
- 3) Query data from Fuseki (and GIT) via an Export API.
- 4) Visual representation of the metadata in a browser.



Status of the 4 parts:

- 1 Work in progress. As of April 28th there is an untested alpha version that is capable of importing JSON data into Fuseki. This part of the code is written in JAVA.³ Other parts of the import functionality are written in R.
- 2 Importing of other data (i.e. not NONMEM or R) is desired. This is however not scope of this project. One way to address that issue is to provide ONE API LAYER that does support any tool generating output on one side and both access to the file and the metadata store on the other side. According to the existing information it is not clear, if there is such an API defined today.⁴
- 3 Export API – this is not defined at the moment, but this component is crucial to the success of the workflow project.
- 4 Some components of the browser interface are defined and will be delivered by scinteco until the end of the project.

³ Mail from Gareth Smith, April 28th

⁴ Mail from Andrew Dunn, April 21th

Timeline

Proposal sent to DDMoRe workflow leaders	May 9 th
Decision from DDMoRe	May 13 th
Specification agreed	May 20 th
Start of Programming	May 23 rd
Phase 1 Visual Prototype (w/o Fuseki Integration)	July 4 th
Phase 2 Fuseki integration	July 29 th
Phase 3 Additional Functions & Documentation	August 26 th
Delivery	August 26 th

Workflow Visualization – User Interface

To provide an impression of the intended look and feel sample screen shots are provided.

Login

When starting the web application a login dialog is displayed.

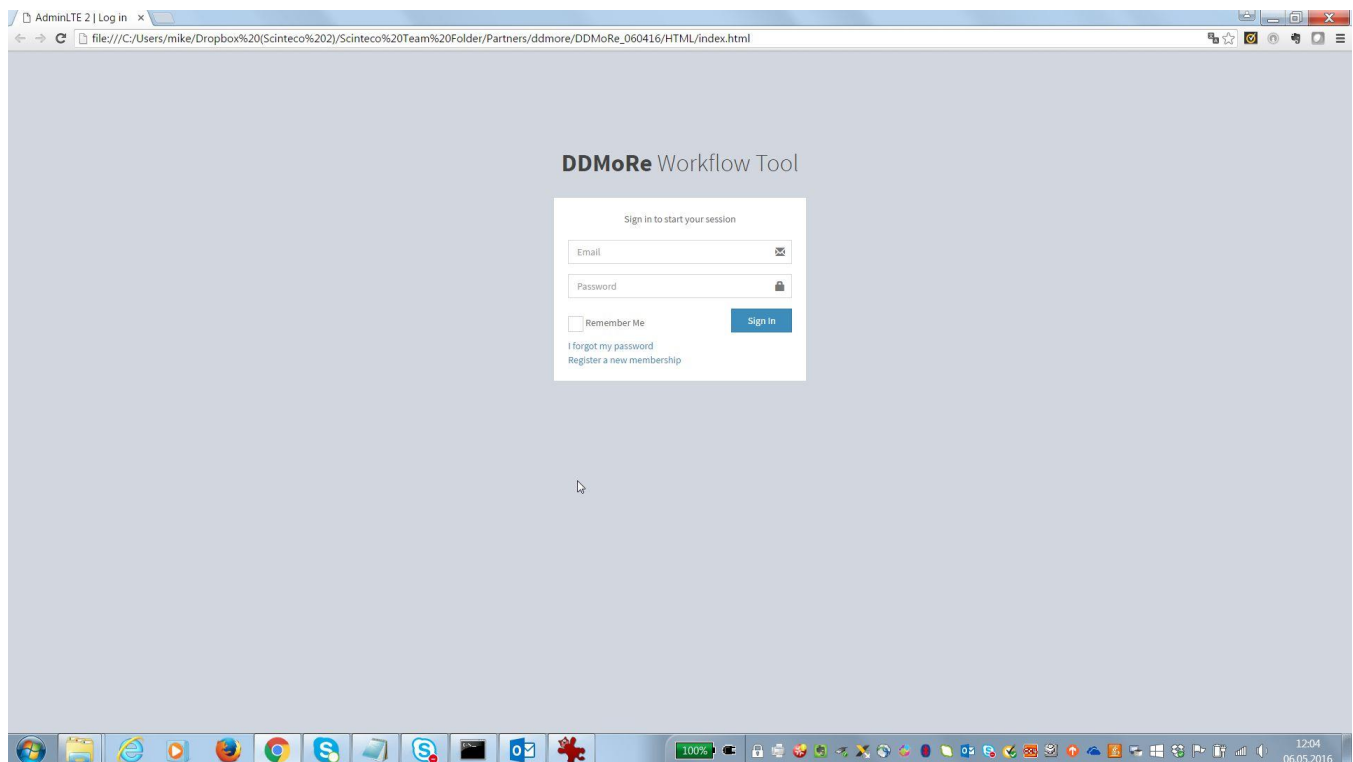
If user management is configured then this user is checked against a central user directory. If not, the default user "ddmore" with password "2016" can log in.

The user may request the reset of his password. He receives an email with instructions and a link to perform this action.

A new user can register here. Either a system administrator must allow access (moderated) or the system administrator is just informed via email about the new user. The new user must specify:

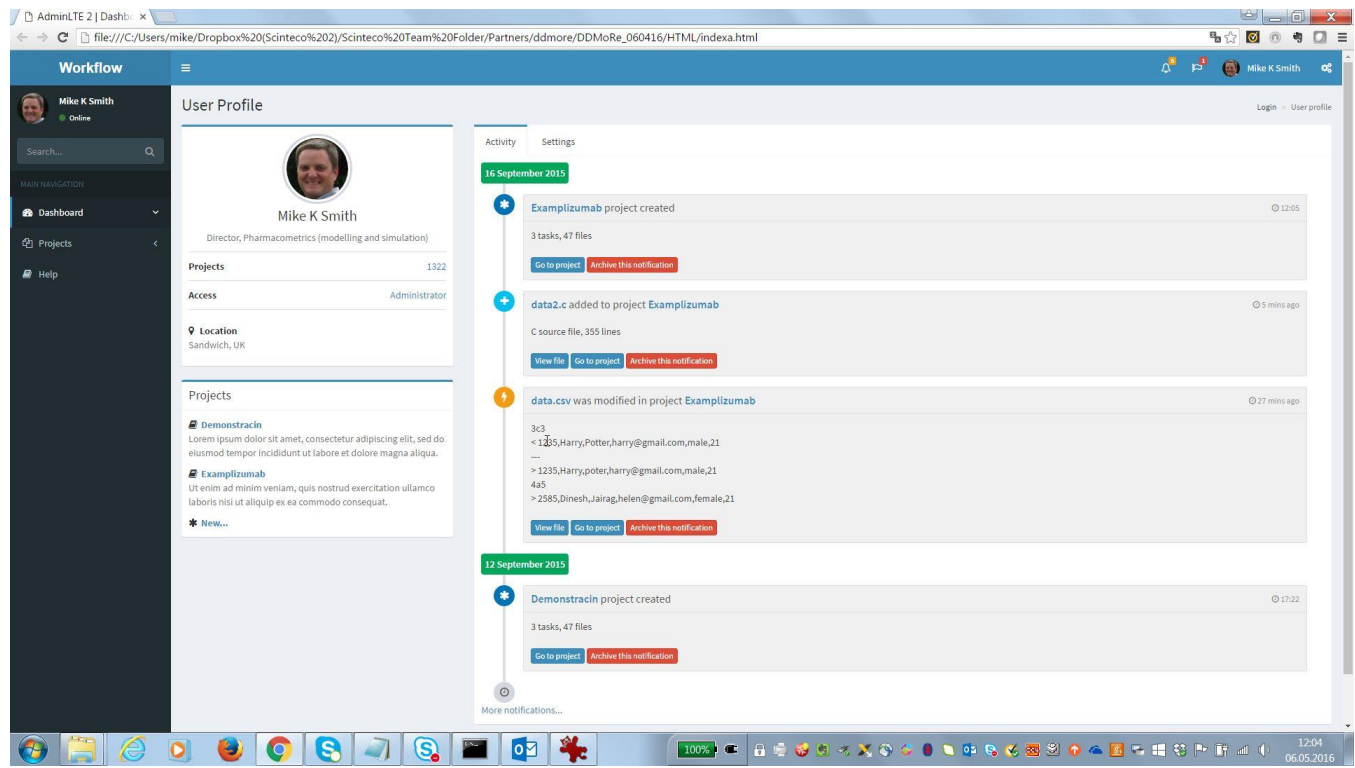
- Email
- Password
- First name
- Last name
- Company (optional)

The user receives an email and must click on the link to activate his account.



After login: news

After login the user receives a screen like this (dashboard view)



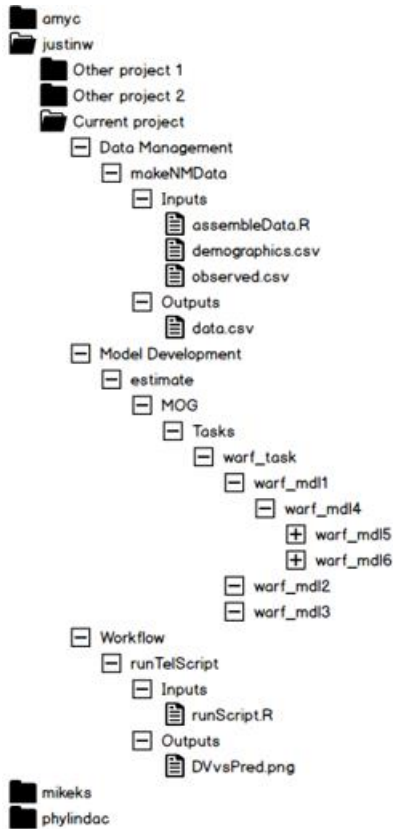
The user can change his personal settings, including the upload of a profile picture.

By clicking on "projects" on the left-hand menu, the user sees a list of all his projects, or of all project, if he is a manager.⁵

⁵ This security concept needs to be detailed (e.g. supporting groups) at a later point in time.

Projects

Displays a list of projects:



The hierarchy is as follows (file view):

User

Project 1

Project 2

Project x

Data Management

Inputs....

Outputs...

Model Management

Estimate..

A click on any project provides a view on a specific project:

Project visualization

Looking at a single project you see 3 main parts:

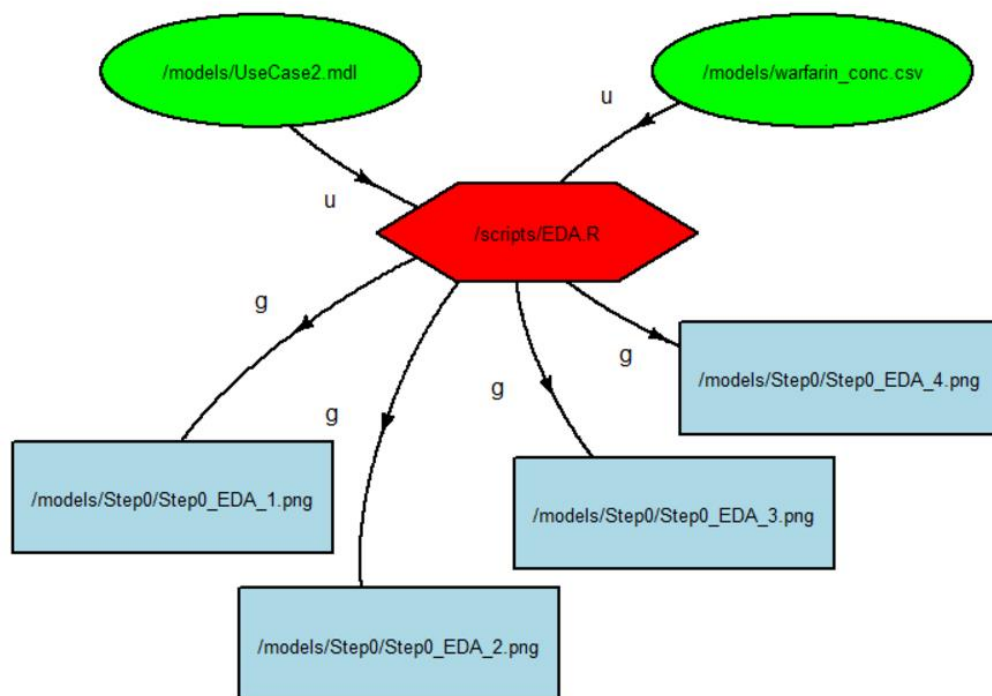
- Input data
- One or more estimate models (in this case warfarin coded in MDL)
- Output data (e.g. graph created with R)
-

The screenshot displays the DDMoRe Workflow 'Thoughtflow' user interface. The main window shows a project titled 'Exemplizumab' with a 'Rerun: assembleData.R' action. Below this, there are three main sections: 'makeNMDData', 'Estimate', and 'runTelScript'. The 'makeNMDData' section contains a single object 'assembleData.R' with a sub-object 'data.csv'. The 'Estimate' section contains a tree structure of objects: 'warf_md1' (with sub-objects 'warf_md4', 'warf_md5', and 'warf_md6'), 'warf_md2', and 'warf_md3'. The 'runTelScript' section contains a single object 'runScript.R' with a sub-object 'DVsPred.png'. At the bottom of the main content area, there are 'Start' and 'Cancel' buttons. On the right side, there is a 'Summary' section with placeholder text and an 'Edit...' button, followed by an 'Options' section with buttons for 'Add object...', 'Export...', 'Re-run', 'History...', and 'Clone...'. Below that is a 'Details' section showing the 'Owner' as 'Mike K Smith' and a list of users with their roles: 'Phylinda Chan' (RO), 'Lutz Harnisch' (RO), and 'Mike K Smith' (RW). The bottom status bar shows the last change: 'warf_md6 by Mike K Smith at 11:01 CET 16.09.2015'. The top navigation bar includes a search bar and a 'Main Navigation' menu with 'Dashboard', 'Projects', and 'Help' options. The bottom taskbar shows various application icons and the system clock indicating 12:04 on 06.05.2016.

Step visualization

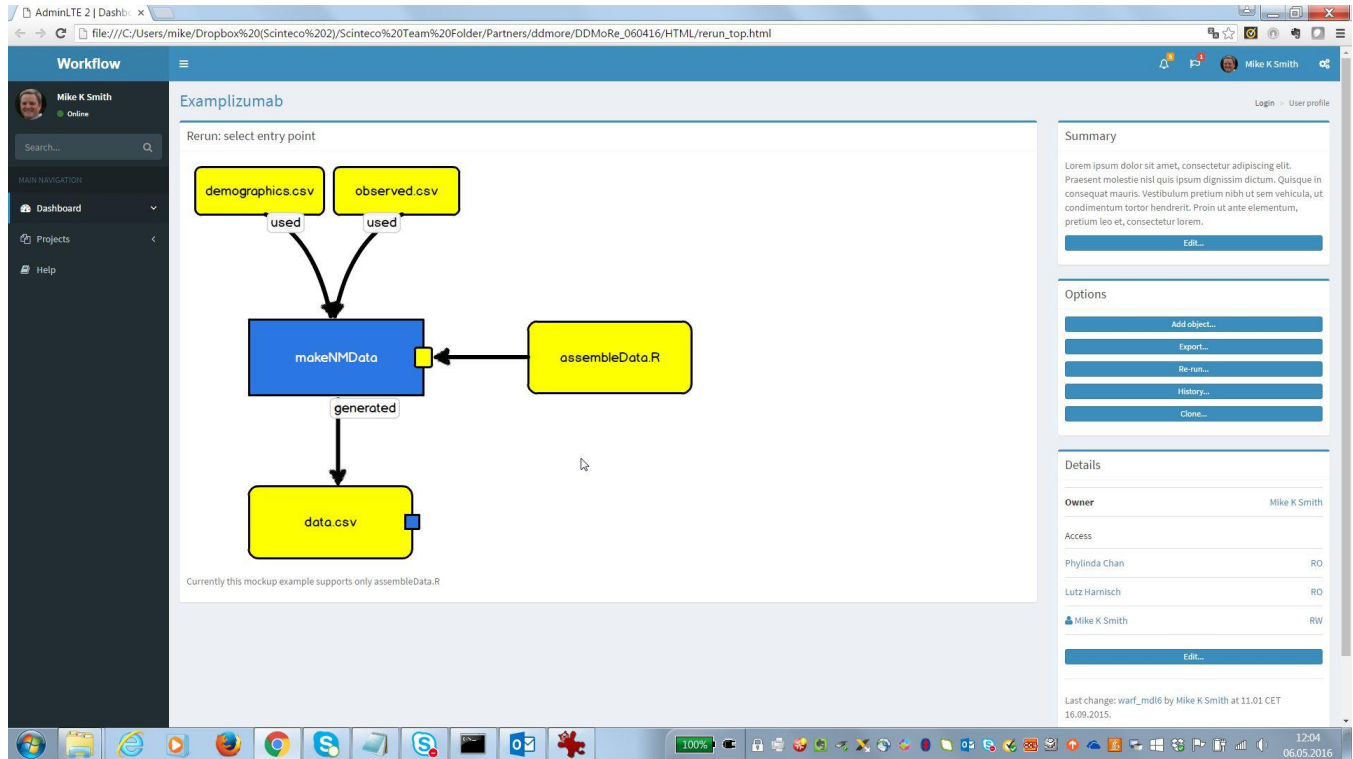
A graphical representation could look like this:

The input files (warfarin concentration as a CSV-file and the script (in this case MDL)) are represented by green ovals, the R script executed is represented with a red hexagon and the 4 resulting diagrams with blue rectangles. The RDF nomenclature is used to design input (u = used) and output (g = generated).



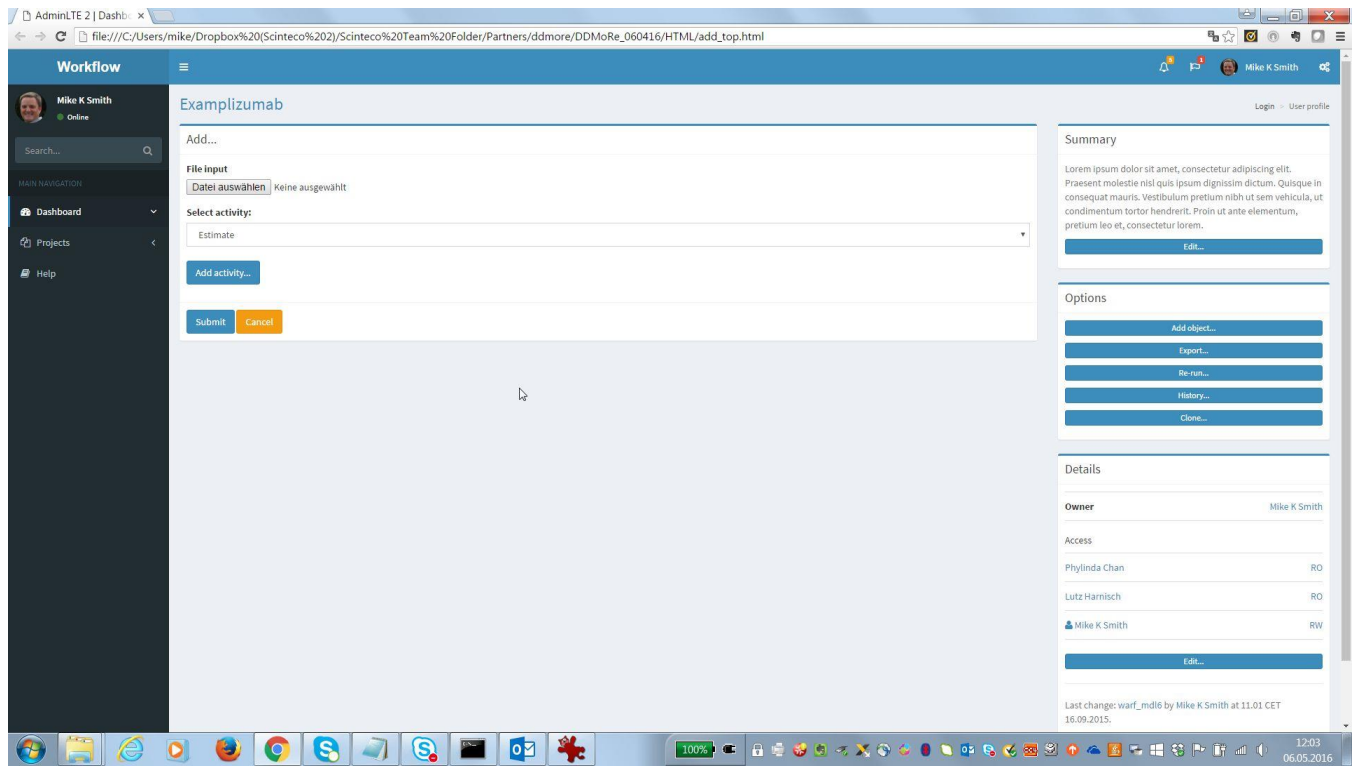
Re-Run

The user can select (e.g. graphically or by selecting files in a file-like view) which input data and which model should be used for re-execution.

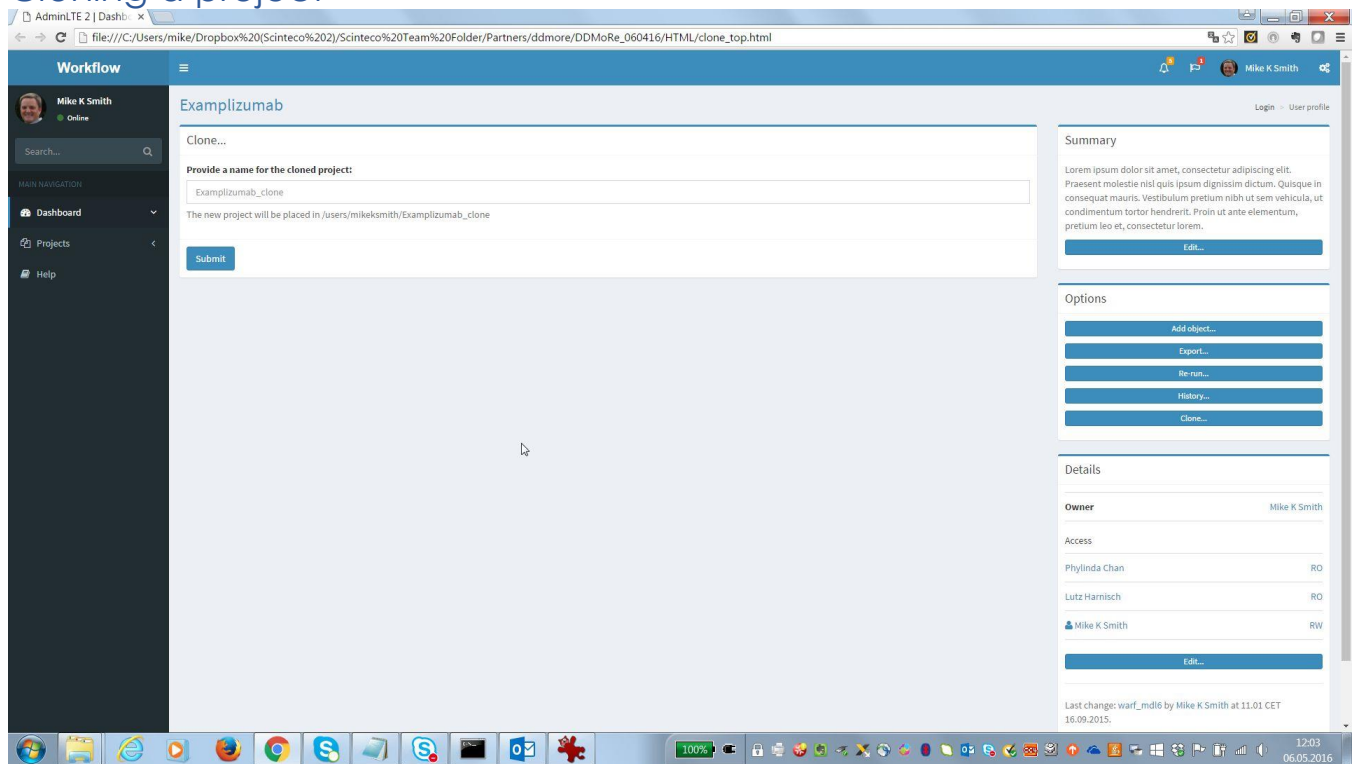


Importing a file

Files (e.g. scripts for estimates) can be imported:



Cloning a project



Specify a project that is to be copied.

Provide a new name. It is placed under the current user's home directory.

DDMoRe Workflow "Thoughtflow"

...

Edit a script

The screenshot displays the DDMoRe Workflow "Thoughtflow" user interface. The main window is titled "Exemplizumab" and shows a script editor with the following content:

```
TEL_Exemplizumab_141015.R

# Guided-demo: Exploring TEL functionality with the warfarin model
# -----
# UseCase2
# -----

# Initialisation
# =====
# Clear workspace
rm(list=ls(all=T))

# Set working directory
uc<-"UseCase2"
wd<-getwd()
setwd(paste("D:/SEE-1.2-SNAPSHOT-30SEP2015/IDL_IDE/workspace/",uc,sep="/"))

# List files available in working directory
list.files()

# Introduction to DDMoRe, TEL
# =====

# Set name of .mdl file
mdlfile <- paste0(uc,".mdl")

# View objects within the .mdl file using TEL
# -----
# Use TEL function getModelObjects() to retrieve model object(s)
# from an existing .mdl file
myMDLobj <- getModelObjects(mdlfile)
length(myMDLobj)
names(myMDLobj)
```

The right sidebar contains a "Summary" section with a placeholder text and an "Edit..." button. Below it is an "Options" section with buttons for "Add object...", "Export...", "Re-run...", "History...", and "Clone...". The "Details" section shows the "Owner" as Mike K Smith and a table of "Access" for three users: Phylinda Chan (RO), Lutz Harnisch (RO), and Mike K Smith (RW). The "Last change" is noted as "warf_mdl6 by Mike K Smith at 11:01 CET 16.09.2015".



Suggested Team Roles

Justin Wilkins	Occams	Project Manager
Jonathan Chard	Mango	Development Import functionality
NN	NN	Development Export API
Mike Beer	scinteco	Subproject Management
NN	scinteco	Development Web Application
Christian Flandorfer	scinteco	Consulting
Michael Hackl	scinteco	Architect, specialist for semantic databases

Contingency Considerations

Risk Factors

There are several risk factors in this project. While the timeframe for completing a solution and the budget available are quite small any issues that could endanger the completion date (August 2016) need to be addressed.

The following risk factors should be considered:

- 1 Provenance database Fuseki – the decision to use this technology seems to be quite new and the supporting interfaces are not available. The import is currently in an untested "alpha" condition, the export API is not available at all.
- 2 Import API not ready, does only support R and NONMEM.
- 3 Export API not even specified.
- 4 Does a User Management for the whole DDMoRe system exist?

Risk Number	Propability of occurence	Impact
1	M	H
2	M	H
3	H	H
4	M	L

So there are 3 classes of risk:

- 1) Critical: not having access to data in the provenance database (export API) RISK #3
- 2) Severe: Usage of Database Fuseki #1 and getting data into Fuseki/GIT #2
- 3) No user management

Potential actions:

For (3) an LDAP-based user management with a web interface can be written by scinteco. This will be an optional part of the proposal. Other DDMoRe-components would also have access to this data.

➔ scinteco can supply this component.

(2) seem to work partly, however the interface between the input (messages generated automatically by a framework) and the fuseki database is not fully documented.

➔ Review the specification with an architect from scinteco

For (1) – Export API – urgent action is necessary.

⇒ Specification by an architect from scinteco

⇒ Programming by scinteco

This adds the following contingency work packages:

TFWP10C user management (JAVA, LDAP)

TFWP11C review import function

TFWP12C specification of export function

TFWP13C creation of export API (JAVA)

Required Proposal Items from scinteco

TFWP7 BWF – Phase 1

This package consists of a browser interface in responsive design (PC, mobile, tablet) written in JAVA/JAVASCRIPT using tomcat and/or apache. In this phase DDMoRe supplies Data (in JSON format) to simulate access to the apache fuseki database.

The user needs to log in, but central user management can be configured (on/off). If it is not available, the user "ddmore" with password "2016" can enter the system.

After login the user sees the "news" screen – giving information on any recent changes in the database.

The user may search in the database (full text search) and receives a hit list.

Clicking on the hit list provides metadata information (e.g. run record). By clicking on a file, the file is opened for display. In this phase this is read only.

Data is also displayed graphically (providing a view of input and output files).

Users with the attribute "manager" can also see a list of projects going on (DASHBOARD VIEW).

TFWP8 BWF – Phase 2

This package integrates the browser application with the fuseki database and git to provide read access to the system. If user management is available by that time, it needs to be integrated.

TFWP9 BWF – Phase 3

The browser application is updated to write to fuseki and git. New functions:

Invite for review: a user can select an item (e.g. file or directory) and invite people for review. This is done by entering email addresses (one or more – if more separated by "," or ";" or " ").

Reviewers can look at entities (e.g. models) and may enter metadata: e.g. a comment or set a flag like "QC passed" or "peer reviewed". Each entry is stored with date/time and user into the fuseki database.

Users can set metadata (attributes) of a model:

- No designation
- Base model
- Final model

Additionally scripts can be edited and stored. There must be versioning using the git-database.

If it is possible to trigger execution of a model from outside (i.e. NOT FROM THE DDMORE ECLIPSE ENVIRONMENT) and a documentation is available, script execution will be considered.

Scripts can be executed and rerun. A new version is stored in git, metadata (who did what and when) is stored in fuseki.

TFWP10C

user management (JAVA, LDAP) – just email, firstname, lastname, company and “manager”-flag. Upload of a profile picture should be supported. Self registration of users with confirmation email, info mail to sys admin. The sys admin can lock or delete users, reset passwords, edit data. Export and import users as CSV.

TFWP11C

review import function: JSON import into Fuseki, GIT-integration issues.

TFWP12C

specification of export function out of fuseki and GIT

TFWP13C

creation of export API (JAVA) – based on TFWP12C.

Open Issues

- 1 User Management: is there a central user management or should BWF provide a user management of its own? This could be an LDAP-like database to be used by all DDMoRe components/users.
- 2 Import API: is there a defined import API to bring data into GIT and Fuseki? How does this work from a NONMEM/R perspective today?
- 3 For some functions (e.g. edit script, re-run a script) we need to be able to write to GIT and fuseki. We have no documentation on this.
- 4 When a new user registers, is it necessary that the sys admin activates the user? Or does he just get the information via email that there is a new user?
- 5 Re-Run: can the execution of a model be triggered from a web server? Or is execution currently only possible out of the DDMoRe/Eclipse/R environment? If this is the case then integrating re-runs will be beyond the scope of this project.

Appendix

Mail from Gareth Smith (April 28th 2016)

Re: Accessing Test Data for Provenance

<https://www.dropbox.com/s/cb1tumlb63rva5f/workspace-prov-dist-20160419.zip?dl=0>

Above is an source provenance data store.

You should be able to download the zip directly.

The ZIP contains a provenance data store test server (Tomcat) and client project to import data into the data store. The reference data is 36 provenance JSON documents, created by the DDMoRe workflow design group. The JSON is representative of the type of messages sent by the DDMoRe provenance logging layer.

All the example server and client code is Java based.

We only did a release of the server 1 week ago so the server code should be regarded as an untested "alpha".

To run the server and build the source code, you will need a Java 7 or 8 installation and Eclipse. There are notes on how to run the test software in the file 'README.txt'.

I wrote the notes for DDMoRe developer, we share a standard platform so the notes are somewhat brief.

I can supply further text as needed.

The example data is in the folder 'client-test\src\test\resources\20160401_provjson'.

Best Regards,

Gareth



Mail from Andrew Dunn (April 21st 2016)

- Are we locked into git for workflow? I think ware, right? (As opposed to svn or anything else.)

Justin Wilkins, PhD

The answer is in the short term yes, but in the long term no. The services/messaging etc are designed to be agnostic of the repository however there has to be some implementation in place and what we have now are services that understand Git and do stuff with Git repositories, however putting SVN versions in at a later date is possible in the design.

Appendix: DDMoRe Project Work Packages⁶

WP1	Model Library Prototypes
WP2	Interoperability Framework
WP3	Modelling Description Language
WP4	Modelling Markup Language / Ontologies
WP5	Integration of Existing Software
WP6	Integration of New Software
WP7	Public Instance of the Modelling Framework
WP8	Communication and Documentation
WP9	Education and Training
WP10	Project Management

⁶ <http://www.ddmore.eu/content/what-do-workpackages-do>