



Model qualification procedure

Author(s):	Siv Jonsson, Wendy Aartsen, Stephan Benay, Roberto Bizotto, Marylore Chenel, Meindert Danhof, Lutz Harnisch, Charlotte Kloft, , Christian Laveille, Paolo Magni, Andrea Mari, Alain Munafo, Zinnia Parra-Guillen, Maria-Luisa Sardu, Celine Sarr, Inaki Troconiz
Date:	24 May 2016
Reviewer(s):	
Approved by:	

Table of contents

1	Introduction	4
2	Objectives	4
2.1	Overall objectives and principles of the MQP	4
2.2	Objectives of the present document	5
3	The model qualification procedure	5
3.1	Models eligible for model certification	5
3.2	Purpose of the MQP	6
3.3	Model Review Group	6
3.4	Future extensions	7
4	The model review process	7
4.1	Workflow of review	7
4.2	Requirements/further documentation	8
4.3	The review process	12
5	Appendix I: Minimal requirements for publication in the DDMoRe Model Repository	13
5.1	Submission guidelines	13
6	Appendix II: Template for request of review for model certification	24
7	Appendix III: Template for the reviewer’s assessment for model certification	27

Abbreviations

DDMoRe	The Drug Disease Model Resources
ExC	Executive Committee
IMI	Innovative Medicines Initiative
MID3	Model Informed Drug Discovery and Development
MDL	Model Description Language
MRG	Modelling Review Group
MQP	Model Qualification Procedure
PharmML	Pharmacometrics Markup Language
PD	pharmacodynamic
PK	pharmacokinetic

1 Introduction

The DDMoRe Model Repository is an open, public, free-to-use platform to store models relevant for Model Informed Drug Discovery and Development (MID3). DDMoRe relies on the scientific community to build knowledge and fill the repository with models used in MID3, spanning from pharmacokinetic (PK) to pharmacodynamic (PD) and disease models and from empirical to systems pharmacology/biology models.

The DDMoRe Model Repository includes specific features for models represented by the Pharmacometrics Markup Language (PharmML). Together with DDMoRe's unique Interoperability Framework, these tools facilitate the collaborative development of computational models. Users can submit their models in various formats, but to benefit from all the features of the model repository the use of the PharmML representation, possibly obtained via the human-readable Model Description Language (MDL) format via the Interoperability Framework tools, is necessary.

Models are published in the DDMoRe Model Repository if fulfilling specified minimal requirements that are automatically checked when uploading the model. However, to ensure that models published in the DDMoRe Model Repository can be trustfully used in MID3, DDMoRe has envisaged a Model Qualification Procedure (MQP). The MQP involves a review process that assesses the completeness and accuracy of the model documentation and its technical validity, as described in this document. A model that is qualified as technically valid receives a DDMoRe model certification, which will be displayed in the model repository. Once finalized, the MQP will be implemented on show-case models publicly available in the DDMoRe Model Repository.

2 Objectives

2.1 Overall objectives and principles of the MQP

Model documentation available on the DDMoRe Model Repository has gone through an automatic check upon submission, to verify that the minimal requirements for publication in the DDMoRe Model Repository are fulfilled (see Appendix I: Minimal requirements for publication in the DDMoRe Model Repository), i.e. verifying that the required files, file descriptions and annotations are present. However, this automatic check does not guarantee (via an independent review) that the model is executable and corresponds to the paper describing the model. In addition, the implemented automatic checking procedure cannot verify if the material submitted to describe the model (metadata, auxiliary files) is appropriate and complete.

The objective of the MQP is to ensure

- that a model in the repository associated with the *DDMoRe model certification* is appropriately and completely documented, can be executed using the Interoperability Framework tools and is able to reproduce the key findings reported in the original associated publication.

The MQP is a human-based process that is performed only if the model submitter requests the DDMoRe model certification. In this case, the model submission must comply with rules that extend the minimal requirements for model publication, which are necessary to make model validation possible. The reviewer that evaluates the model performs all the necessary validation steps and provides conclusions about the model validity. The DDMoRe model certification is granted if a model is considered valid, i.e. is successfully passing the MQP. The certification is associated to a specific version of a model and if a new version of the model is uploaded, then a new certification is required. A DDMoRe certified model may be safely downloaded from the repository and used for the purposes for which it was proposed.

The current MQP only considers the validity of a model from a technical point of view, i.e. does not evaluate the appropriateness of the model for any use e.g. in the drug development process or research. Nevertheless, the technical validity ensures that a model is executable and corresponds to what is reported in the publication.

Currently, the MQP is envisaged only for a subclass of the models that can be uploaded to the repository, as detailed in this document. Future extensions are possible, as outlined in section 3.4 Future extensions.

2.2 Objectives of the present document

This document describes the context, objectives, requirements and process of the MQP, as it is envisaged at this stage of the DDMoRe project.

3 The model qualification procedure

The MQP concerns models published in the DDMoRe Model Repository and, therefore, it presupposes that the requirements for DDMoRe model publication are met. These minimal requirements comprise of requirements concerning annotation of models (annotation that is mandatory is defined in the Metadata end user requirements (last version as of May 24th, 5.9), as well as requirements concerning the documentation needed to be provided. The requirements enforced concerning the documentation are dependent on the type of model code submitted (i.e. the submission scenario) as described in the Submission Guidelines, which is defined Appendix I: Minimal requirements for publication in the DDMoRe Model Repository, and displayed on the Model Repository website (<http://repository.ddmore.eu/>).

Models eligible for the MQP (see section 3.1 Models eligible for model certification) in general need further documentation (detailed in section 4.2 Requirements/further documentation) in addition to the minimal requirements, to prove the technical validity of the model.

The MQP is based on the following general principles:

- The model must be publicly available on the DDMoRe Model Repository and certification must be requested by the model submitter.
- To obtain certification, the model must meet eligibility criteria for the MQP (see section 3.1 Models eligible for model certification).
- The MQP involves a model review process that is carried out by a body of qualified experts (see section 3.3 Model Review Group), who evaluates the technical model validity.
- The model review and evaluation is concluded by a decision of granting or rejecting the certification, possibly after further interactions with the model submitter.
- The DDMoRe model certification is displayed on the repository, with a short report of the model review leading to the decision of acceptance, thereby describing the model characteristics that have been reviewed and is the basis for the model certification.

3.1 Models eligible for model certification

In addition to the minimal requirements necessary for model submission and publication in the DDMoRe Model Repository, the prerequisites for the MQP with respect to the model are the following:

- The model is published in full-length peer-reviewed article and sufficiently described.
- The model is coded in executable MDL/PharmML.

- The MDL/PharmML model code represents the original model without modifications or simplifications that may alter its properties.
- The model submitter has access to the results from the model using the real original data. (Only applicable to models intended for estimation, i.e. not applicable for a simulation model.)

3.2 Purpose of the MQP

As anticipated above, the MQP concerns only the technical aspects of the model. The purpose of the MQP is to:

- Check that the information associated with the model on the model repository (metadata, auxiliary documentation files) is provided, makes sense and is sufficient to understand the model.
- Check that the submitted model code represents the original model in the associated publication without modifications that may alter its properties.
- Check that the command file allows execution of the model using the Interoperability Framework and includes tasks that are sufficient to prove that the submitted model corresponds to the associated publication.
- Decide if the documentation provided supports that the model in the repository is the same as the one described in the publication associated to the model.

A model is considered technically valid, and can thus be granted the model certification, if all the above steps are passed.

The assessment of the technical validity is made by reviewers, as detailed below. Guidelines for the model review are provided, see section 4.3 The review process, and to facilitate and make the assessment of the model uniform, a pre-specified checklist/template is used in the model review process (see Appendix III: Template for the reviewer's assessment for model certification).

3.3 Model Review Group

The experts entitled to perform the review according to the MQP constitute the Model Review Group (MRG). The MRG will cover expertise in the modelling and simulation area with specific experience in the DDMoRe framework.

Two members of the MRG will be chosen for the review of each model, whereof one performs the review and the second is available for second opinion. Experts external to the MRG may be nominated if necessary.

The chosen reviewers should have no Conflict of Interest for the purpose of critically reviewing a proposed model. No Conflict of Interest means:

- Should not be submitter or co-author of the publication associated to the model.
- If the model is submitted on behalf of a company, the reviewer should be from a public institution.

Reviewers are asked to declare that they do not have relevant conflicts of interest and to refuse the invitation to review in case of conflicts.

3.4 Future extensions

The principles underlying the current MQP may be extended in the future to expand the classes of models admitted to the MQP. Extensions will involve different requirements and evaluation rules. The following priority is envisaged for extensions:

- Models not published in full-length in peer-reviewed journals (e.g. conference abstracts) or unpublished models.
- Models coded in the original target code for which it is not possible to obtain an executable MDL/PharmML code due to limitations of these coding languages and/or the language converters.
- Models for which the submitter does not have access to results from the real original data used in the associated publication (in particular relevant historical models).

Broader aspects of model certification may also be considered in the future, for example:

- Model certification from a regulatory point of view: In addition to the technical model certification, the review and certification may support the regulatory process, by considering requests from regulatory authorities.
- Model certification from a scientific journal point of view: In addition to the technical model certification, the review and certification may support the journal submission process, by considering requests from scientific journals.

These alternative certification aspects will require further consultancy with regulatory agencies and scientific journals.

4 The model review process

4.1 Workflow of review

To manage the review process a Review supervisor will be nominated by the Model Review Group. The Review supervisor will act as an intermediate between the model submitter and the reviewer(s) and will manage potential controversies between the model submitter and the reviewer(s). The model review workflow follows the scheme illustrated in Figure 1. In particular:

1. The model review for model certification is requested by the submitter.
 - Documentation, in addition to minimal requirements, is provided according to a pre-specified template (see Appendix II: Template for request of review for model certification) and acknowledged by the Review supervisor.
2. Reviewers are identified and appointed by the Review supervisor.
3. Review process
 - One reviewer assesses the technical validity of the model according to a pre-specified checklist/template (see Appendix II: Template for request of review for model certification).
 - The reviewers discuss the assessment, where the second reviewer serves as second opinion, agreement between reviewers is reached, the first review report is produced and sent to the Review supervisor.
 - The first review report is communicated to the submitter by the Review supervisor with one of the following outcome:

compatible with the Interoperability Framework, i.e. is written in MDL/PharmML and is executable. When the model was originally submitted for publication in the DDMoRe repository a script for execution of the model using the Interoperability Framework environment was required, i.e. the command file (see Appendix I: Minimal requirements for publication in the DDMoRe Model Repository). This command file can be used as the basis for the validation plan. The validation plan, should in addition describe the steps undertaken by the submitter to demonstrate the adherence of the proposed model implementation to the published one (e.g reproduction of parameter estimates and precision, simulations). Thus, the purpose of the validation plan is that it should include the commands required to reproduce in a complete way selected (main) tasks described in the paper (e.g. simulations, parameter estimates and precision). Compared to the command file required for model publication, the MQP requires that:

1. the validation plan includes execution commands to perform the tasks that are necessary to prove convincingly that the repository model corresponds to the publication.
2. the validation plan contains a clear explanation of the tasks performed.

Additional plots, documentation etc. required to conclude the adherence of the proposed model implementation to the published one, i.e. to conclude the validation plan, may be provided as found necessary by the submitter.

Furthermore, additional *output files* following execution of the model are requested as detailed in Table 1. These requirements depend on whether the model is intended for estimation or simulation, and if real original or simulated data is used in the validation plan. Please note, if the validation plan is performed with real original data not available in the repository (e.g human data), the submitter should be in the position to share them with the reviewer to reproduce the validation plan.

Table 1 Additional files to provide for model certification in various situations.

Model task	Data provided	Additional files to provide for model certification	Reason for requirement
Estimation	Real original data, published in model repository (animal) or privately shared (human)	<ol style="list-style-type: none"> 1. Validation plan 2. <i>Optional:</i> Output file from the original analysis code with estimation results obtained using the real original data. The output file must include the objective function value obtained with the final parameter estimates. 3. <i>Optional:</i> The original model code 	<p>For the reviewer to be able conclude that the model code is executable and to conclude adherence of the proposed model implementation to the published one</p> <ol style="list-style-type: none"> 1. by running the commands included in the validation plan using the real original data to produce parameter estimates and uncertainty, plots etc. and compare them with the publication. <p>The consistency of the estimation results will be evaluated against those in the paper (and those obtained from the original analysis code if provided (optional))</p>
Estimation	Simulated data published in model repository, real original data not shared	<ol style="list-style-type: none"> 1. Validation plan 2. Document describing the features of the simulated data and how the simulated data was generated (i.e. is it a dummy dependent variable or is it simulated from the model, how were covariates included, etc), however, a complete script is not required). 3. Output file using the executable MDL/PharmML code with estimation results obtained using the real original data. 4. When simulated data contain a dummy dependent variable: Output file from the original analysis code with estimation results obtained from the simulated data. 	<p>For the reviewer to be able conclude that the model code is executable and to conclude adherence of the proposed model implementation to the published one</p> <ol style="list-style-type: none"> 1. by comparing provided output files (parameter estimates and uncertainty etc) using the executable MDL/PharmML code and real original data with those in the publication. <p>The consistency of the estimation results provided in the output files using the real original data with those reported in the publication will be evaluated.</p> <ol style="list-style-type: none"> 2. by running the commands included in the validation plan using the simulated data to produce parameter estimates and uncertainty, plots etc. and compare them with the publication.

		<p>5. <i>Optional:</i> Output file from the original analysis with estimation results obtained using the real original data. The output file must include the objective function value obtained with the final parameter estimates.</p> <p>6. <i>Optional:</i> The original model code</p>	<p><i>When the simulated dependent variable is generated from the model, i.e. not a dummy dependent variable:</i> The consistency of the estimation results obtained from simulated data will be evaluated against those obtained from provided output related to the execution of MDL/PharmML code with the real original data and those in the associated publication.</p> <p><i>When the simulated data contain a dummy dependent variable:</i> The consistency of the estimation results obtained from simulated data will be evaluated against those obtained in the provided output related to the original analysis code with estimation results obtained from the simulated data.</p> <p>3. by comparing the translated model code obtained in the output files using the executable MDL/PharmML code and real original data with the translated model code obtained in the output with simulated data</p> <p>The consistency of these two translated model codes will be evaluated.</p>
Simulation	None	1. Validation plan	<p>For the reviewer to be able conclude that the model code is executable and to conclude adherence of the proposed model implementation to the published one</p> <p>1. by running the commands included in the validation plan to produce figures and values and compare with the publication.</p>

4.3 The review process

The model review will be performed by one reviewer and will follow a pre-specified checklist/template, see Appendix III: Template for the reviewer's assessment for model certification).

The review will initially check that the submitter has provided all the requested documentation and that the documentation is accurate and appropriate.

Thereafter, a technical review will take place evaluating: model eligibility, validation plan, model execution, model parameters and uncertainty, model simulations, additional documentation provided, according to a list of questions in the template.

Based on the technical review, an assessment is made whether the provided documentation (model and validation plan) supports that the model in the repository is in agreement with the one described in the publication associated with the model.

Finally, the reviewer summarizes what has been tested to demonstrate the adherence of the proposed model implementation to the one described in the associated publication and, if relevant, what is considered to be further needed to demonstrate the adherence of the proposed model implementation to the one described in the related publication. Thus, a conclusion is made whether

- the model is accepted for model certification, i.e. the model documentation is complete and accurate, the model code is executable, and the model is described and coded in agreement with the associated publication.

or

- further information is required before final decision.

In the latter case a list of questions is included in the review report.

5 Appendix I: Minimal requirements for publication in the DDMoRe Model Repository

For publication of a model in the DDMoRe model repository, certain minimal requirements need to be fulfilled with respect to

- annotation of the model
- documentation to be uploaded

Both these parts comprise the minimal requirements for publication. The purpose of these requirements is that uploaded models should be searchable and executable.

The model annotation required refers to the metadata identified as mandatory for publication of a model, according to Metadata end user requirements, version 5.9. The annotation is included during submission of the model with an annotation tool in the DDMoRe Model Repository. In a transition phase (while the annotation tool is developed) among all the annotation indicated as mandatory in the Metadata end user requirements, version 5.9, only those actually available in the annotation tool are required.

The requirements enforced concerning the documentation to be uploaded are dependent on the type of model code submitted (i.e. the submission scenario) as described below in section 5.1 Submission guidelines.

The fulfilment of the minimal requirements is checked automatically without human intervention. Thus, there are no requirements per se of the scientific content, but DDMoRe relies on submitters to fill the DDMoRe Model Repository with relevant content.

5.1 Submission guidelines

You are about to submit a new model to the DDMoRe Repository. The scientific community thanks you!

The following submission guidelines have been defined to guide you through the necessary steps (minimal requirements) you have to follow to bring your model into the public domain with high quality information. Please read them carefully.

Please Note: If you encounter issues preventing you from publishing your model in the DDMoRe Repository, you can share your model with the Modelling Review Group (MRG) (click on your model entry then on the share button on the left). In doing so, you will benefit from their expertise and receive hints on how to improve your model's compliance with the minimal requirements.

Ground submission rules

- The quality of the information depends solely on submitters.
- Any model can be published if fulfilling the minimum requirements (described in the next steps).
- If a model refers to a publication it has to have the same or equivalent mathematical representation. If this is not the case, the publication can still be referred to but

modifications to the publication and their motivations need to be clearly detailed in the uploaded documents.

- Uploading of observed human data is not accepted. This might change in the future as the project evolves.
- A simulated data set is needed to let other users:
 - understand how the data set has to be constructed, i.e. which data items, specific records, etc., are needed;
 - run the model, e.g. execute estimation/simulation OR execute an evaluation (MAX=0 in NONMEM): the type of execution is the submitter's decision.

Therefore, a data set with the dependent variable simulated from the uploaded model OR with a dummy dependent variable is acceptable.

- Output files are needed to
 - prove the model is executable (output from execution of simulated data is sufficient for this purpose);
 - report the final model parameter estimates and corresponding uncertainty (output from execution of real data is needed for this purpose).

When real data are accessible to the submitter, the output file obtained from the real data needs to be provided. When real data are not accessible, an output file containing the information about parameter estimates and their uncertainty, as available in the publication, needs to be provided.

Terms of agreement

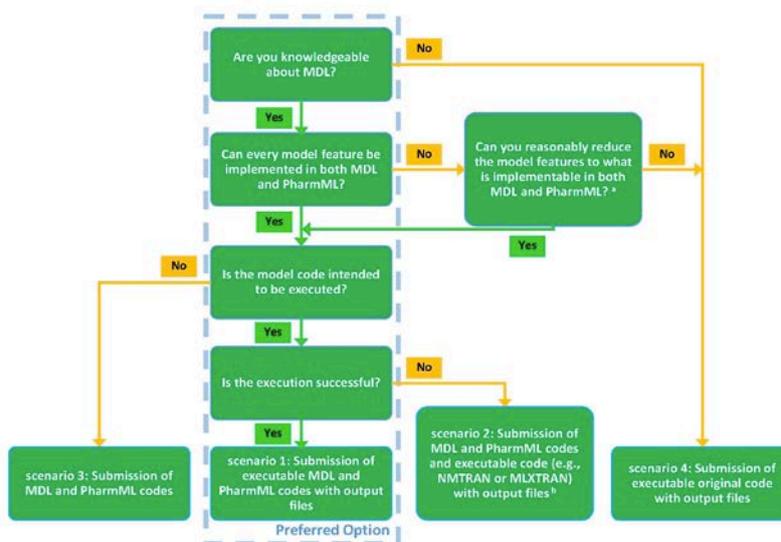
- You are about to submit a new model to the DDMoRe Repository.
- Submissions are linked to a [User Name] identified by login information.
- Login information should be accurate and not entered on behalf of someone else.
- It is understood that every deposition is the responsibility of the submitter. Once the submitter presses the 'Publish' button, the submitter accepts to share the model, the uploaded data, and any additional file with the public. The submitter is the only responsible for deciding to make the model, uploaded data and additional files publicly available. If the information provided were not already in public domain, obtaining the approval of co-authors/contributors is the responsibility of the submitter.
- DDMoRe takes no responsibility for the accuracy of the information provided.
- By publishing or sharing a model, the submitter agrees on the terms above.

Create relevant login credentials

If not already registered to the DDMoRe Repository, register on <http://repository.ddmore.eu/>. Please be aware that your name will be displayed as submitter for the models you upload, thus define it appropriately: first name, space, last name (you can also modify it once logged in selecting 'Edit User' from your profile page).

Identify the submission scenario

To allow as many models as possible to be uploaded, a number of submission scenarios have been identified based on the possibility to generate MDL and/or PharmML code. The documentation required to upload a model varies based on the scenario chosen. Therefore, please use the decision tree below to identify which case is applicable to your model. Once that is done, follow carefully the instructions below for the chosen scenario, all steps are mandatory if not differently specified.



^a The option to upload a model with reduced functionality is provided to cater for situations where the MDL and/or PharmML language versions supported by the Repository do not support all the features of the original model. However, a companion file named *Model_Accommodations.txt* needs to clearly state how the model was modified in order to implement it in MDL and PharmML.

^b The option to upload MDL and PharmML codes which are not executable is provided to cater for situations where the automatic converters from these languages to target tools languages are not aligned with the most recent developments of MDL and/or PharmML. Modifications to the code produced by automatic conversion from MDL or PharmML need to be described in the companion file named *Command.txt*.

Follow the file naming rules

Before preparing the material to be uploaded to the DDMoRe Repository, please look at the naming rules for the files to be uploaded [below](#).

Go to [scenario 1](#) | [scenario 2](#) | [scenario 3](#) | [scenario 4](#)

Scenario 1: Executable MDL and PharmML codes

This applies when the original model can be successfully coded in both MDL and PharmML and that the MDL and PharmML codes can be executed in the DDMoRe Interoperability Framework.

Example model: Friberg_2009_Schizophrenia_Asenapine_PANSS
 [http://repository.ddmore.eu/model/DDMODEL00000002]

1. Start submission by uploading PharmML code (version up to 0.6.1) as mandatory file (the DDMoRe Repository will thus provide an automatic human-friendly model visualization within the Model Definition tab).
2. Upload the following additional files. Add a short and meaningful description in the box next to each file.
 - a) MDL code (version 7.0 or later supported by DDMoRe Interoperability Framework 4.1 or later).
 - b) Real data (possible if non-human) or simulated data. If simulated: provide a dataset similar to the real dataset with dependent variable(s) simulated from the uploaded model OR with dummy dependent variable(s), and with covariates replaced by their typical values.
 - c) Command.txt, which describes the successful execution command/script: this file also needs to specify the uploaded file names (for input data, executable model, and output results of the execution) and the used language/product versions/target tool, see [example 1](#).
 - d) Output file obtained after execution of uploaded dataset and MDL model code.
 - e) Additional output file (if you decided to upload simulated data), with the final model parameter estimates and the corresponding uncertainty. If you have access to real data and do not upload them, this is the output file obtained after execution with real data. If you do not have access to real data but the model refers to a publication, this file should include final estimates and corresponding uncertainty as given in the publication used as reference, see [output example](#).
 - f) Model Accommodations.txt if you had to modify the published model: describe in this file the differences between the implemented model and the model in the publication used as reference. Use [this template](#) and see also examples in the following two model entries: Hamren_2008_diabetes_tesaglitazar [<http://repository.ddmore.eu/model/DDMODEL00000003>], and Trefz_2015_metabolism_Kuvan_TurnoverKPD [<http://repository.ddmore.eu/model/DDMODEL00000103>].
 - g) Any other files considered to be useful.
3. After pressing Upload, fill in Model Information.

In the box 'Name': change the default to a name describing your model (max 150 characters). This is the name which will be shown when browsing the DDMoRe Repository (see [<http://repository.ddmore.eu/models>] for examples).

In the box 'Description': add a brief and meaningful description of your model.
4. Next add a reference for your model. If the model is related to a publication, provide the corresponding publication identifier: PubMed ID (to be preferred), DOI, other link (URL) or choose 'Publication without link'.
5. Before pressing 'Complete Submission', check the 'Summary of your submission' (click on 'Back' within the repository and refine if necessary).

6. After completing the submission, enter the model annotations by pressing the button 'Annotate' (see the buttons in the left bar) and fill in the fields on all available tabs; then press 'Save' to save the information you provided and 'Validate' to check the validity of the annotations you entered.
7. Click on 'Return to model display page' and then on the 'Publish' button (see the buttons in the left bar) in order to share your model with the community and complete the publishing process. Before publishing, please ensure you provided all the files and information described in the previous steps, otherwise the (automatic) publishing process will reject the model publication. If you need to modify your model to have it published, click on the button 'Update' in the left bar.

Scenario 2: MDL and PharmML codes and executable code

This scenario applies when the MDL and PharmML codes cannot be entirely executed in the DDMoRe Interoperability Framework (via translation of MDL or PharmML to the target tool language) and an additional executable code (e.g., NMTRAN or MLXTRAN) must be provided.

Example model: Hamren_2008_diabetes_tesaglitazar
[\[http://repository.ddmore.eu/model/DDMODEL00000003\]](http://repository.ddmore.eu/model/DDMODEL00000003)

1. Start submission by uploading a non-executable PharmML code (version up to 0.6.1) as mandatory file (the Model Repository will thus provide an automatic human-friendly model visualization within the Model Definition tab).
2. Upload the following as additional files. Add a short and meaningful description in the box next to each file.
 - a) A non-executable MDL code (version 7.0 or later supported by DDMoRe Interoperability Framework 4.1 or later).
 - b) An executable code (e.g., NMTRAN or MLXTRAN). The submitter is encouraged to use the MDL or PharmML code, convert it into the target tool language using the DDMoRe Interoperability Framework, and modify the produced code in order to make it executable.
 - c) Real data (possible if non-human) or simulated data. If simulated: provide a dataset similar to the real dataset with dependent variable(s) simulated from the uploaded model OR with dummy dependent variable(s), and with covariates replaced by their typical values.
 - d) Command.txt, which describes the successful execution command/script: this file also needs to specify the uploaded file names (for input data, executable model, and output results of the execution) and the used language/product versions/target tool. Moreover, the modifications mentioned in point b) above need to be described within this file: as an example, see Hamren_2008_diabetes_tesaglitazar [\[http://repository.ddmore.eu/model/DDMODEL00000003\]](http://repository.ddmore.eu/model/DDMODEL00000003).
 - e) Output file obtained after execution of uploaded dataset and executable code.
 - f) Additional output file (if you decided to upload simulated data), with the final model parameter estimates and the corresponding uncertainty. If you have access to real data this is the output file obtained after execution with real data. If you do not have access to real data and do not upload them, this file should include final estimates and corresponding uncertainty as given in the publication used as reference, see [output example](#).

- g) Model Accommodations.txt if you had to modify the published model: describe in this file the differences between the implemented model and the model in the publication used as reference (any changes that were required in the executable code to make it run should not be stated here but in the Command.txt file). Use [this template](#) and see also examples in the following two model entries: Hamren_2008_diabetes_tesaglitazar [<http://repository.ddmore.eu/model/DDMODEL00000003>], and Trefz_2015_metabolism_Kuvan_TurnoverKPD [<http://repository.ddmore.eu/model/DDMODEL00000103>].
 - h) Any other files considered to be useful.
3. After pressing Upload, fill in Model Information.
In the box 'Name': change the default to a name describing your model (max 150 characters). This is the name which will be shown when browsing the DDMoRe Repository (see [<http://repository.ddmore.eu/models>] for examples).
In the box 'Description': add a brief and meaningful description of your model.
 4. Next add a reference for your model. If the model is related to a publication, provide the corresponding publication identifier: PubMed ID (to be preferred), DOI, other link (URL) or choose 'Publication without link'.
 5. Before pressing 'Complete Submission', check the 'Summary of your submission' (click on 'Back' within the repository and refine if necessary).
 6. After completing the submission, enter the model annotations by pressing the button 'Annotate' (see the buttons in the left bar) and fill in the fields on all available tabs; then press 'Save' to save the information you provided and 'Validate' to check the validity of the annotations you entered.
 7. Click on 'Return to model display page' and then on the 'Publish' button (see the buttons in the left bar) in order to share your model with the community and complete the publishing process. Before publishing, please ensure you provided all the files and information described in the previous steps, otherwise the (automatic) publishing process will reject the model application. If you need to modify your model to have it published, click on the button 'Update' in the left bar.

Scenario 3: MDL and PharmML codes, not intended for execution

This scenario applies when the model is not intended to be executed in the DDMoRe Interoperability Framework.

Example model: None currently

1. Start submission by uploading a non-executable PharmML code (version up to 0.6.1) as mandatory file (the Model Repository will thus provide an automatic human-friendly model visualization within the Model Definition tab).
2. Upload the following additional files. Add a short and meaningful description in the box next to each file.
 - a) A non-executable MDL code (version 7.0 or later supported by DDMoRe Interoperability Framework 4.1 or later)
 - b) Model Accommodations.txt if you had to modify the published model: describe in this file the differences between the implemented model and the model in the publication used as reference. Use [this template](#) and see also examples in the

following two model entries: Hamren_2008_diabetes_tesaglitazar
[\[http://repository.ddmore.eu/model/DDMODEL00000003\]](http://repository.ddmore.eu/model/DDMODEL00000003), and
 Trefz_2015_metabolism_Kuvan_TurnoverKPD
[\[http://repository.ddmore.eu/model/DDMODEL00000103\]](http://repository.ddmore.eu/model/DDMODEL00000103).

- c) Any other files considered to be useful.
3. After pressing Upload, fill in Model Information.
 In the box 'Name': change the default to a name describing your model (max 150 characters). This is the name which will be shown when browsing the DDMoRe Repository (see [\[http://repository.ddmore.eu/models\]](http://repository.ddmore.eu/models) for examples).
 In the box 'Description': add a brief and meaningful description of your model.
 4. Next add a reference for your model. If the model is related to a publication, provide the corresponding publication identifier: PubMed ID (To be preferred), DOI, other link (URL) or choose 'Publication without link'.
 5. Before pressing 'Complete Submission', check the 'Summary of your submission' (click on 'Back' within the repository and refine if necessary).
 6. After completing the submission, enter the model annotations by pressing the button 'Annotate' (see the buttons in the left bar) and fill in the fields on all available tabs; then press 'Save' to save the information you provided and 'Validate' to check the validity of the annotations you entered.
 7. Click on 'Return to model display page' and then on the 'Publish' button (see the buttons in the left bar) in order to share your model with the community and complete the publishing process. Before publishing, please ensure you provided all the files and information described in the previous steps, otherwise the (automatic) publishing process will reject the model publication. If you need to modify your model to have it published, click on the button 'Update' in the left bar.

Scenario 4: Executable original code

This scenario applies when the model submitter wants to publish a model that for various reasons has not been coded in MDL and PharmML.

Example model: CMS_Colistin_PK_Karaiskos_2015
[\[http://repository.ddmore.eu/model/DDMODEL00000130\]](http://repository.ddmore.eu/model/DDMODEL00000130)

1. Start submission by uploading executable original code as the mandatory file (thus, the DDMoRe Repository will NOT be able to provide an automatic human-friendly model visualization).
2. Upload the following additional files. Add a short and meaningful description in the box next to each file.
 - a) Real data (possible if non-human) or simulated data: provide a dataset similar to the real dataset with dependent variable(s) simulated from the uploaded model OR with dummy dependent variable(s), and with covariates replaced by their typical values.
 - b) Command.txt, which describes the successful execution command/script: this file also needs to specify the uploaded file names (for input data, executable model, and output results of the execution) and the used target tool, see [example 2](#).
 - c) Output file obtained after execution of uploaded dataset and executable original model code (it is possible to use equivalent to MAX=0 from NONMEM to save time if simulated dataset is used).

- d) Additional output file (if you decided to upload simulated data), with the final model parameter estimates and the corresponding uncertainty. If you have access to real data this is the output file obtained after execution with real data. If you do not have access to real data, this file should include final estimates and corresponding uncertainty as given in the publication used as reference, see [output example](#).
 - e) Any other files considered to be useful.
3. After pressing Upload, fill in Model Information.
 In the box 'Name': change the default to a name describing your model (max 150 characters). This is the name which will be shown when browsing the repository (see <http://repository.ddmore.eu/models> for examples).
 In the box 'Description': add a brief and meaningful description of your model.
4. Next add a reference for your model. If the model is related to a publication, provide the corresponding publication identifier: PubMed ID (to be preferred), DOI, other link (URL) or choose 'Publication without link'.
5. Before pressing 'Complete Submission', check the 'Summary of your submission' (click on 'Back' within the repository and refine if necessary).
6. After completing the submission, enter the model annotations by pressing the button 'Annotate' (see the buttons in the left bar) and fill in the fields on all available tabs; then press 'Save' to save the information you provided and 'Validate' to check the validity of the annotations you entered.
8. Click on 'Return to model display page' and then on the 'Publish' button (see the buttons in the left bar) in order to share your model with the community and complete the publishing process. Before publishing, please ensure you provided all the files and information described in the previous steps, otherwise the (automatic) publishing process will reject the model publication. If you need to modify your model to have it published, click on the button 'Update' in the left bar.

Naming rules for all uploaded files

Note that in the given names below the '_xxx' string stands for the free part of the file names. Exact spelling (case-sensitive) needs to be respected.

- Data
 - **Real_xxx** — real, non-human
 - **Simulated_xxx** — simulated
- Executable model code
 - **Executable_xxx**. The executable file is a file which can be executed. With scenario 1, at least two files are executable: the MDL and the PharmML codes. With scenario 2, this name cannot be used for the MDL and PharmML codes. Make sure that the name of the input data in the model code is in agreement with the data file you upload.
- Non-executable model code
 - All files which are not executable must not start with Executable_xxx. In particular: for scenario 2 and 3, model codes written in MDL or PharmML cannot be named Executable_xxx.

- Execution command/script
 - **Command.txt**. Make sure that the names of the input data, executable code and output file in the script are in agreement with the files you upload.
- Output file
 - **Output_real_xxx** — from execution using real data. If real data are not uploaded, this is the additional output file introduced above, i.e. a file with the final model parameter estimates and the corresponding uncertainty obtained either from publication or from execution with real data.
 - **Output_simulated_xxx** — from execution using simulated data. Make sure that, if shown, the names of the input data and executable code in this file are in agreement with the files you upload.
- Modifications with respect to the published model
 - **Model_Accomodations.txt** — text file describing how the uploaded mandatory model differs from the related publication referred to.

Go to [scenario decision tree](#)

Files content examples

Command.txt — example 1

Follow this example for the mandatory Command.txt file if scenario 1 applies.

It is highly recommended to follow this example with the initial block of commented lines filled in.

```
##### Scenario = 1
##### MDL version = <MDL version, constrained to be 7.0 at the
moment>
##### IO product = <interoperability product number, constrained to
be 4.1 at the moment>
##### Input data =<file name of uploaded input data>
##### Executable model = <file name of uploaded executable>
##### Output = <file name of uploaded output from executable>

#Initialisation
rm(list=ls(all=F)) #clean your workspace first

#Set working directory
myfolder<- "" # name of your project folder
setwd(file.path(Sys.getenv("MDLIDE_WORKSPACE_HOME"),myfolder))

#Set name of .mdl file and dataset for future tasks
mymodel<- ""
datafile<- ""
mdlfile<- paste0(mymodel, ".mdl")

#ESTIMATE model parameters using Monolix
mlx <- estimate(mdlfile, target="MONOLIX", subfolder="Monolix")

# Print the estimated parameters
```

```
parameters_mlx<- getPopulationParameters(mlx, what="estimates")$MLE
print(mlx)

# Perform model diagnostics for the base model using Xpose functions
mlx.xpdb<-as.xpdb(mlx,datafile)

pdf("GOF_MLX.pdf")
print(basic.gof(mlx.xpdb))
print(ind.plots(mlx.xpdb))
dev.off()
# This code can be expanded upon users needs to e.g. include
execution in other tools
```

Go to [scenario 1](#) | [scenario 2](#) | [scenario 3](#) | [scenario 4](#)

Command.txt — example 2

Follow this example for the mandatory Command.txt file if scenario 4 applies. In this specific case the original code is run with NONMEM via PsN.

It is highly recommended to follow this example with the initial block of commented lines filled in.

```
##### Scenario = 4
##### PsN version = <PsN version>
##### NONMEM version = <NONMEM version>
##### Input data =< file name of uploaded input data>
##### Executable model = <file name of uploaded executable>
##### Output = <file name of uploaded output from executable>

execute run1.mod
```

Go to [scenario 1](#) | [scenario 2](#) | [scenario 3](#) | [scenario 4](#)

Output_real_xxx

This is an example for the **Output_real_xxx** file to be used when the submitter has no access to real human data.

It can be a text file or a pdf file with a table, showing the final model parameter estimates and the corresponding uncertainty as provided in the related publication. The text file can be arranged as follows:

Parameter name [as in the model given in publication]	Parameter estimate (unit)	Relative standard error (%) [or Standard error; as
CL	104.5 (L/h)	15%
V	25.4 (L)	23%
etc.		

Go to [scenario 1](#) | [scenario 2](#) | [scenario 3](#) | [scenario 4](#)

Model_Accommodations.txt — example

This file has to be uploaded when the model is not implemented as in the original publication.

```
##### Model differences:
```

```
##### Reasons:
```

Go to [scenario 1](#) | [scenario 2](#) | [scenario 3](#) | [scenario 4](#)

6 Appendix II: Template for request of review for model certification

Question/Document required	Response/Documentation provided For submitter to fill in grey areas
Requested for all models irrespective of task	
Name of model, including version, to review for model certification	[name of model and version as stated in model repository, link to model [http://repository.ddmore.eu/model/DDMODEL####]]
Name of model submitter	[model submitter name]
Confirm that model is coded in executable MDL/PharmML	I confirm that the uploaded model is coded in executable MDL/PharmML: [choose: yes/no]
Confirm that the MDL/PharmML model code represents the original model without modifications or simplifications that may alter its properties as published in the full-length peer-reviewed article	I confirm that the uploaded MDL/PharmML model code is not modified compared with the model published in the full-length peer-reviewed article: [publication information as given in model repository]
<p>Validation plan</p> <p>Validation approach</p> <p>The model is [an estimation/a simulation] model.</p> <p>Validation is achieved by</p> <p>Choose/add other:</p> <ul style="list-style-type: none"> - executing the uploaded executable MDL/PharmML code with the real original data in the DDMoRe framework - executing the uploaded executable MDL/PharmML code with the simulated data in the DDMoRe framework - executing the uploaded original target code with the simulated data - providing a table comparing i) parameter estimates and uncertainty from the original target model code and the real original data with ii) parameter estimates and uncertainty from the uploaded executable MDL/PharmML code and the real original data - Figure xrep, which is a reproduction of Figure x from the related publication by a simulation using the uploaded executable MDL/PharmML and [simulation tool used]. Figure x is provided as reference [name of file]. Add any necessary details - Figure yrep, which is a reproduction of Figure y from the related publication by a simulation using the uploaded executable MDL/PharmML and [simulation tool used]. Figure y is provided as reference [name of file]. Add any necessary details - etc 	

<p>Validation material [detail all files needed to go through this validation plan for example as follows]</p> <ul style="list-style-type: none"> • MDL/PharmML code from the repository [name] • Script used for model execution [name] • Model code/script used for simulation [name] • Reference figure x [name] • Reference figure y [name] • Script X used for generating Figure xrep [name] • Script B used for generating Figure xrep [name] <p>etc</p> <p>Validation steps</p> <ul style="list-style-type: none"> • Download to the working directory the PharmML model and other files provided • Open R and set the working directory to the directory where the PharmML file is located • Run the script name • Run the script name <p>etc</p> <ul style="list-style-type: none"> • Compare the output files from the execution of the executable MDL/PharmML code with the real original data to the same information in the related publication and to the original output files. • Compare the produced Figure xrep to the reference Figure X (name) <p>etc</p> <p>Remark: any additional information for reviewer</p>	
Other documents (optional)	The following additional documents aiming to support the conclusion of the adherence of the proposed model implementation to the published one have been uploaded/provided: [names of files]
Conclusion of model validation	The execution of the validation plan supports that the uploaded model code is executable in the DDMoRe framework, and that the model is described and coded in agreement with the related publication. In particular the following are evidence for this conclusion: <ul style="list-style-type: none"> 1. xxx 2. yyyy <p>etc</p>
Additional request for models intended for estimation - Data	

<p><i>Optional:</i> Provide real original data</p>	<p>The file [name] is the real original data. I confirm that this file is uploaded/provided.</p>
<p><i>If the real original data are not provided:</i> Description of simulated data</p>	<p>The simulated data named [name] have the following features and was generated as follows (i.e. is it a dummy dependent variable or is it simulated from the model, how were covariates included, etc. A complete script is not needed): [describe data items etc here]</p>
<p>Additional request for models intended for estimation – Output files <i>If the real original data are not provided</i></p>	
<p>Model output file(s) including parameter estimates and uncertainty from the uploaded executable MDL/PharmML and the real original data.</p>	<p>The file(s) [name(s)] is/are the model output file(s) from the uploaded executable MDL/PharmML and the real original data. I confirm that this/these file is/are uploaded/provided.</p>
<p>When simulated data contain a dummy dependent variable: Original target model output file(s) using the simulated data and obtained from the execution of a model evaluation or model re-estimation.</p>	<p>The file(s) [name(s)] is/are the original target model output file(s) using the simulated data. I confirm that this/these file(s) is/are uploaded/provided.</p>
<p>Optional for submitter to provide for models intended for estimation <i>If the real original data are not provided</i></p>	
<p><i>Optional:</i> Original target model code</p>	<p>The file [name] is the original target model code. I confirm that this file is uploaded/provided.</p>
<p><i>Optional:</i> Original target model output file(s) including parameter estimates and uncertainty from real original data. The output file must include the objective function value obtained with the final parameter estimates.</p>	<p>The file(s) [name(s)] is/are the original target model output file(s). I confirm that this/these file(s) is/are uploaded/provided.</p>

7 Appendix III: Template for the reviewer’s assessment for model certification

Question/Document required	For reviewer to fill/modify/choose in grey areas
Name of model, including version, to review for model certification	[name of model and version as stated in model repository, link to model [http://repository.ddmore.eu/model/DDMODEL####]
Name of model submitter	[model submitter name]
Related publication in full-length peer-reviewed article	[publication information as given in model repository]
Documentation provided - all models irrespective of task	
Has the submitter completed the “Template for request of review for model certification” and provided the requested documents?	[choose: Yes/No, the submitter has not provided xxx and yyy and the request needs to be completed prior to review as follows: 1. specify 2. specify]
Documentation provided – Data (only applicable for models intended for estimation)	
Is one of the following provided: Description of simulated data or Real original data	[choose: Yes, a description of the simulated data is provided./ Yes, the original real original data are provided./ No, the documentation is incomplete.].
Documentation provided – Output files (only applicable for models intended for estimation) <i>If the real original data are not provided</i>	
Is the model output file(s) including parameter estimates and uncertainty using the uploaded executable MDL/PharmML and the real original data provided?	[choose: Yes/No, the documentation is incomplete.].
When simulated data contain a dummy dependent variable: Is the original target model output file(s) using the simulated data following execution of a model evaluation or model re-estimation provided?	[choose: Yes/No, the documentation is incomplete.].
Additional optional documentation provided	
<i>Optional:</i> Is the original target model code provided?	[choose: Yes/No].

<p><i>Optional:</i> Is the original target model output file(s) including parameter estimates and uncertainty from real original data provided? Is the objective function value obtained with the final parameter estimates?</p>	<p>[choose: Yes/No].</p>
<p>Technical review of model/model eligibility</p>	
<p>Is the model sufficiently described in full-length peer-reviewed article?</p>	<p>[choose: Yes, the model is sufficiently described in full-length peer-reviewed article./ No, the model is not sufficiently described in the full-length peer-reviewed article to be able to make a review for model certification. The model is therefore not eligible for model certification.]</p>
<p>Is the model coded in MDL/PharmML?</p>	<p>[choose: Yes, the model is coded in MDL/PharmML./ No, the model is NOT coded in MDL/PharmML and is therefore not eligible for model certification.]</p>
<p>Does the model represent the original model without modifications or simplifications that may alter its properties?</p>	<p>[choose: Yes, the model corresponds to the one in the related publication./ No, the model does not correspond to the one in the related publication and is therefore not eligible for model certification. The difference refer to xxx, yyy etc.]</p>
<p>Has the model submitter access to results from the real original data? <i>Only applicable to models intended for estimation, i.e. not applicable for a simulation model.</i></p>	<p>[choose: Yes, the model submitter access to results from the real original data./ No, the model submitter has NOT access to results from the real original data and the model is therefore not eligible for model certification./ The model is intended for simulation and this requirement is thus not applicable.]</p>
<p>Technical review of validation plan</p>	
<p>Is the process of model validation clear (adequately and completely described) in its aims and steps?</p>	<p>[choose: Yes./ No, additional information is required to demonstrate the adherence of the proposed model implementation to the one described in the related publication. The additional information/validation step refers to xxx and yyy. (Same information in the List of questions in the end)]</p>
<p>Is the provided/uploaded material complete for the validation plan?</p>	<p>[choose: Yes./ No, additional information is required to demonstrate the adherence of the proposed model implementation to the one described in the related publication. The additional information/validation step refers to xxx and yyy. (Same information in the List of questions in the end)]</p>
<p>Can the validation steps be</p>	<p>[choose: Yes, the validation steps was possible to perform by</p>

<p>performed (using the available tools)?</p>	<p><i>the reviewer./</i></p> <p><i>No, the validation steps was NOT possible to perform due to xxx and yyy. (Same information phrased as question in the List of questions in the end)]</i></p>
<p>Technical review of execution</p>	
<p>Can the uploaded model be executed in the DDMoRe framework?</p>	<p><i>[choose: Yes, the uploaded model could be executed in the DDMoRe framework by the reviewer./</i></p> <p><i>No, the uploaded model could not be executed in the DDMoRe framework due to xxx and yyy. (Same information phrased as question in the List of questions in the end)]</i></p>
<p>Technical review of model output, including for example model code, parameters and uncertainty</p>	
<p><i>Not applicable for simulation models</i></p>	
<p>Does the uploaded model produce parameter estimates and uncertainty from real original data in agreement with the related publication?</p>	<p><i>[choose: Yes, the parameter estimates and their uncertainty following execution of the uploaded executable MDL/PharmML model from real original data by the reviewer are in agreement with those in the related publication./</i></p> <p><i>Yes, the parameter estimates and their uncertainty, based on the submitter provided output files following execution with the uploaded executable MDL/PharmML model with the real original data set, are in agreement with those in the related publication./</i></p> <p><i>No, the parameter estimates and their uncertainty following execution of the uploaded executable MDL/PharmML model with the real original data set by the reviewer are NOT in agreement with those in the related publication./</i></p> <p><i>No, the parameter estimates and their uncertainty, based on the submitter provided output files following execution with the uploaded model with the real original data set, are NOT in agreement with those in the related publication./]</i></p>
<p><i>When the simulated dependent variable are generated from the model, i.e. not a dummy dependent variable:</i> Are the outputs following execution of uploaded executable MDL/PharmML and the simulated data in agreement with those obtained in the related publication?</p>	<p><i>[choose: Yes, the parameter estimates and their uncertainty following execution of the uploaded executable MDL/PharmML model from simulated data by the reviewer are in agreement with those in the related publication./</i></p> <p><i>No, the parameter estimates and their uncertainty following execution of the uploaded executable MDL/PharmML model from simulated data by the reviewer are NOT in agreement with those in the related publication./</i></p>
<p><i>When the simulated data contain a dummy dependent variable:</i> Are the outputs in agreement following execution (evaluation</p>	<p><i>[choose: Yes, the output with respect to choose: parameters and their uncertainty/OFV/predictions/other following execution (choose: evaluation/re-estimation) of the original target model code and the simulated data in the target</i></p>

<p>or re-estimation) of</p> <p>i) the original target model code and the simulated data in the target software.</p> <p>and</p> <p>ii) the corresponding target model code generated from the uploaded executable MDL/PharmML and the simulated data.</p>	<p><i>software, based on the submitter provided output files are in agreement with the output following execution of the corresponding target model code generated from the uploaded executable MDL/PharmML and the simulated data set./</i></p> <p><i>No, the output with respect to choose: parameters and their uncertainty/OFV/predictions/other following execution (choose: evaluation/re-estimation) of the original target model code and the simulated data in the target software, based on the submitter provided output files are in agreement with the output following execution of the corresponding target model code generated from the uploaded executable MDL/PharmML and the simulated data set./</i></p>
<p>Is the translated model code obtained in the output files using the executable MDL/PharmML code and real original data in agreement with the translated model code obtained in the output with simulated data?</p>	<p><i>[choose: Yes, the translated model codes obtained in the output files using the executable MDL/PharmML code and real original data and in the output with simulated data are in agreement./</i></p> <p><i>No, the translated model codes obtained in the output files using the executable MDL/PharmML code and real original data and in the output with simulated data are NOT in agreement./</i></p>
<p>Technical review of simulations, plots and other documentation provided</p>	
<p>Does the uploaded model produce simulations in agreement with the related publication?</p>	<p><i>[choose: Yes, the simulation results based on the uploaded executable MDL/PharmML model were re-generated by the reviewer using the provided scripts and they are in agreement with the corresponding simulations presented in the related publication./</i></p> <p><i>No, the simulations based on the uploaded executable MDL/PharmML model re-generated by the reviewer are NOT in agreement with the corresponding simulations presented in the related publication.]</i></p>
<p>Assessment of the adequacy of the technical review</p>	
<p>Is the provided documentation (model, validation plan etc) sufficient to support that the uploaded model in the repository is executable, and the model is described and coded in agreement with the related publication?</p> <p>If provided documentation is not sufficient, describe in general the insufficiencies, for example if simulations are not in agreement, what are the differences and the consequences?</p>	<p><i>[choose: The provided documentation fully supports that the uploaded model in the repository is executable, and that the model is described and coded in agreement with the related publication. optional: some more sentences being more of an assessing nature, for example: There were some differences in the parameters and their uncertainty between the uploaded model and the model in the related publication. However, these differences were considered minor since xxx./</i></p> <p><i>or</i></p> <p><i>The provided documentation does not fully support that the uploaded model in the repository choose:</i></p> <p><i>is executable/that the model is described in agreement with the related publication /that the model is code in agreement with the related publication. The insufficiencies refer to: choose</i></p> <p><i>incomplete validation plan as follows: xxx is needed to fully certify the model technically. optional: some more sentences</i></p>

	<p><i>being more of an assessing nature. /</i></p> <p><i>difficulties executing the uploaded model in the DDMoRe framework without modifying it as follows: xxx This needs to be resolved to fully certify the model technically. optional: some more sentences being more of an assessing nature. /</i></p> <p><i>differences in the parameters and their uncertainty between the uploaded model and the model in the related publication These differences are not considered minor and may indicate that the uploaded model differ from the model in the related publication. optional: some more sentences being more of an assessing nature. /</i></p> <p><i>differences in simulated output between the uploaded model and the corresponding output in the related publication. These differences are not considered minor and may indicate that the uploaded model differ from the model in the related publication. optional: some more sentences being more of an assessing nature.</i></p> <p><i>The identified insufficiencies should be addressed by the model submitter, see List of questions.]]</i></p>
<p>Has any important flaw been identified in the review, which may undermine the validity of the model?</p>	<p><i>[The following important flaw has been identified in the review: xxx. This may undermine the validity of the model and should be addressed by the model submitter, see List of questions.]</i></p>
<p>Summary of the submitted documentation and conclusion of review</p>	
<p>If the model will be accepted for model certification:</p> <p>Summarize what has been tested to demonstrate that the uploaded model in the repository is executable, and the model is described and coded in agreement with the related publication.</p>	<p><i>[The submitter has demonstrated that the uploaded model in the repository is executable, and that the model is described and coded in agreement with the related publication as follows:</i></p> <ol style="list-style-type: none"> <i>1. Demonstration that the model is executable in the DDMoRe framework, which was also re-checked by the reviewer. State how this was demonstrated; estimation with real original data or with simulated data etc..]</i> <i>2. Demonstration of agreement between parameters and their uncertainty following execution of the uploaded model using the real original data compared with the same data from the related publication. State the details how this was demonstrated, i.e. by output files generated by the reviewer with uploaded model and real original data or output files provided by the submitter.]</i> <i>3. Demonstration of agreement between simulations from the uploaded model and those presented in the related publication. Specify the actual Figure numbers in the publication and that scripts were provided and re-generated by the reviewer.</i> <i>4. Demonstration of ...]</i>
<p>If a final decision cannot be made at this point:</p> <p>Summarize what is considered to</p>	<p><i>[The submitter has partially provided information to demonstrate that the uploaded model in the repository is executable, and that the model is described and coded in</i></p>

<p>be further needed to demonstrate that the uploaded model in the repository is executable, and the model is described and coded in agreement with the related publication.</p>	<p><i>agreement with the related publication. However, there are parts that have not been fully covered related to the demonstration of choose: model execution/agreement between parameters and their uncertainty following execution of the uploaded model using the real original data compared with the same data from the related publication/agreement between simulations from the uploaded model and those presented in the related publication/other xxxx.</i></p>
<p>Conclusion of review</p>	<p><i>The conclusion of the review is that the uploaded model named [name of model] choose:</i></p> <p><i>Is accepted for model certification, i.e. the model code is executable, and the model is described and coded in agreement with the related publication.</i></p> <p><i>or</i></p> <p><i>Cannot be accepted for model certification currently. Further information is required before a final decision, see List of questions.</i></p>
<p>List of questions to submitter</p>	
<p>Add questions in relation to assessment in the review report</p>	<p>The following questions should be addressed by the model submitter before a final decision concerning model certification can be made:</p> <ol style="list-style-type: none"> 1. Please provide further evidence that.... 2.