# Description of UseCases models in MDL

A set of UseCases (UC) has been prepared to illustrate how different modelling features can be implemented in the Model Definition Language or MDL. These UseCases are located in the “models” folder under the UseCasesDemo project pre-configured in the IDE. The key characteristics represented in each UC can be found in Table I. Please note that development is still ongoing. Some models are not expected to be fully interoperable (estimation is only possible with NONMEM). Some other Use Cases are not included in this release, but are presented here to give an overview of models which will become available in later releases. In addition, a more detailed description of the working UCs is available on the subsequent pages.

Table I: Brief description of UseCases

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Dataset** | **Description** | **Interoperable** | **Included in this release** |
| UC1 | warfarin\_conc.csv | PK model, ODE, single oral administration | YES | YES |
| UC2 | warfarin\_conc.csv | PK model, analytical solution | YES | YES |
| UC3 | warfarin\_conc\_pca.csv | PK and PD outcomes, use of DVID | YES | YES |
| UC4 | warfarin\_infusion\_oral.csv | Different dosing routes with ODE, use of CMT | YES | YES |
| UC4\_1 | warfarin\_infusion\_oral.csv | Different dosing routes with COMPARTMENTS | YES | YES |
| UC5 | warfarin\_conc\_sexf.csv | Categorical covariate and covariate transformations | NO | YES |
| UC6 | warfarin\_conc.csv | PK model, correlation between random effects | YES | YES |
| UC7 | warfarin\_conc\_cmt.csv | PK model (1CMT) with COMPARTMENTS | YES | YES |
| UC8 | warfarin\_conc\_bov\_P4 \_sort.csv | PK model, between occasion variability | NO | YES |
| UC9 | warfarin\_infusion.csv | PK model, IV infusion | YES | YES |
| UC10 | warfarin\_conc\_cmt.csv | PK model (2CMT) using COMPARTMENTS | NO | NO |
| UC10\_1 | warfarin\_conc\_cmt.csv | As UC10 with different parameterisation | NO | NO |
| UC11 | count.csv | Poisson count data | YES | YES |
| UC12 | binary.csv | Binary outcome data, Bernoulli | NO | NO |
| UC12\_1 | binary.csv | Binary outcome data, Binomial | NO | NO |
| UC13 | category.csv | Categorical outcome data | NO | NO |
| UC14 | warfarin\_TTE\_exact.csv | Time to event data, right censoring and exact | YES | YES |
| UC14\_1 | warfarin\_TTE\_ intervalCensored.csv | Time to event data, interval censored | NO | NO |
| UC14\_2 | warfarin\_RTTE\_ intervalCensored.csv | Repeated time to event data | NO | NO |
| UC15 | warfarin\_conc\_cmt.csv | Complex PK model using COMPARTMENT, multiple dosing routes | NO | NO |
| UC16 | BIOMARKER\_simDATA.csv | Multiple observations, log-transformed outcomes, ODE | NO | NO |
| UC17 | warfarin\_conc\_SS.csv | Steady state dosing using SS | NO | YES |
| UC17\_1 | warfarin\_conc\_SS.csv | Steady state dosing using SS, II, ADDL | NO | YES |

## UseCase1

Warfarin population pharmacokinetic model using ordinary differential equations (ODEs)

Dosing regimen: single oral administration

Dataset:

* ID : Patient identifier (n=32)
* TIME [h]
* WT : Patient’s weight [kg]
* AMT : Total drug administered [mg]
* DVID : dependent variable identifier (0: dose,1:PK measurement)
* DV : Warfarin concentration [mg/L]
* MDV : missing dependent variable (0: observation, 1: dosing record)
* logtKG : log transformed patient’s body weight standardised to 70 kg

Structural model:

* 1 compartment model using ODE (V, CL, ka, TLAG)
* 1st order absorption process with lag time
* 1st order elimination process

Covariate model:

* WT on CL and V following allometric principles

Variability model:

Inter-individual variability:

* + - Exponential model for V, CL, ka and TLAG (this last fix to 0.1) expressed as standard deviation
    - Correlation between CL and V random variables expressed in correlation scale

Residual error model:

* Combined error model

## UseCase2

Warfarin population pharmacokinetic model using analytical solutions

Dosing regimen: single oral administration

Dataset:

* ID : Patient identifier (n=32)
* TIME [h]
* WT : Patient’s weight [kg]
* AMT : Total drug administered [mg]
* DVID : dependent variable identifier (0: dose,1:PK measurement)
* DV : Warfarin concentration [mg/L]
* MDV : missing dependent variable (0: observation, 1: dosing record)
* logtKG : log transformed patient’s body weight standardised to 70 kg

Structural model:

* 1 compartment model using ODE (V, CL, ka, TLAG)
* 1st order absorption process with lag time
* 1st order elimination process

Covariate model:

* WT on CL and V following allometric principles

Variability model:

Inter-individual variability:

* + - Exponential model for V, CL, ka and TLAG expressed as standard deviation
    - Correlation between CL and V random variables expressed in correlation scale

Residual error model:

* Combined error model

## UseCase3

Population pharmacokinetic and pharmacodynamic model to describe warfarin and PCA response

Dosing regimen: single oral administration

Dataset:

* ID : Patient identifier (n=32)
* TIME [h]
* WT : Patient’s weight [kg]
* AGE [years]
* SEX (0: female, 1: male)
* AMT : Total drug administered [mg]
* DVID : dependent variable identifier (0: dose,1:PK measurement, 2: PD measurement)
* DV : Warfarin concentration [mg/L] or PCA measurement
* MDV :missing dependent variable (0: observation, 1: dosing record)
* logtKG ; log transformed patient’s body weight standardised to 70 kg

Structural model:

PK model

* 1 compartment model using ODE (V, CL, ka, TLAG)
* 1st order absorption process with lag time
* 1st order elimination process

PD model

* Indirect response model
* 0 order synthesis (RCPA) and 1st order elimination (KPCA)
* Inhibitory effect of drug concentration on RPCA (synthesis) using an Emax model (EMAX, C50)

Covariate model:

* WT on CL and V following allometric principles

Variability model:

Inter-individual variability:

* + - Exponential model for V, CL, ka, TLAG (this last fix to 0.1), PCA0, C50 and TEQ (ln(2)/KPCA) expressed as standard deviation
    - Linear model for EMAX
    - Correlation between CL and V random variables expressed in correlation scale

Residual error model:

* Combined error model for warfarin
* Additive error model for PCA

## UseCase4

Warfarin population pharmacokinetic model for multiple dosing via different administration routes

Dosing regimen: intravenous infusion followed by oral administration

Dataset:

* ID : Patient identifier (n=32)
* TIME [h]
* WT : Patient’s weight [kg]
* AMT : Total drug administered [mg]
* RATE : infusion rate [mg/h]
* CMT : compartment number (1: absorption compartment, 2: central compartment)
* DV : Warfarin concentration [mg/L]
* logtKG : log transformed patient’s body weight standardised to 70 kg
* MDV : missing dependent variable (0: observation, 1: dosing record)

Structural model:

* 1 compartment model using ODE (V, CL, ka, TLAG, FORAL)
* 1st order absorption process with lag time and bioavailability for oral administration
* 0 order input for intravenous infusion
* 1st order elimination process

Covariate model:

* WT on CL and V following allometric principles

Variability model:

Inter-individual variability:

* + - Exponential model for V, CL, ka and TLAG (this last fix to 0.1) expressed as standard deviation
* Logit model for FORAL expressed as standard deviation
* Correlation between CL and V random variables expressed in correlation scale

Residual error model:

* Combined error model

## UseCase5

Warfarin population pharmacokinetic model incorporating categorical and transformed covariates

Dosing regimen: single oral administration

Dataset:

* ID : Patient identifier (n=32)
* TIME [h]
* WT : Patient’s weight [kg]
* AGE [years]
* SEXF (0: male, 1: female)
* AMT : Total drug administered [mg]
* DVID : dependent variable identifier (0: dose,1:PK measurement)
* DV : Warfarin concentration [mg/L]
* MDV :missing dependent variable (0: observation, 1: dosing record)

Structural model:

* 1 compartment model using ODE (V, CL, ka, TLAG)
* 1st order absorption process with lag time
* 1st order elimination process

Covariate model:

* WT on CL and V following allometric principles
* SEX and AGE on CL with an exponential model

Variability model:

Inter-individual variability:

* + - Exponential model for V, CL, ka and TLAG (this last fix to 0.1) expressed as standard deviation
    - Correlation between CL and V random variables expressed in correlation scale

Residual error model:

* Combined error model

## UseCase6

Warfarin population pharmacokinetic model

Dosing regimen: single oral administration

Dataset:

* ID : Patient identifier (n=32)
* TIME [h]
* WT : Patient’s weight [kg]
* AMT : Total drug administered [mg]
* DVID : dependent variable identifier (0: dose,1:PK measurement)
* DV : Warfarin concentration [mg/L]
* MDV : missing dependent variable (0: observation, 1: dosing record)
* logtKG : log transformed patient’s body weight standardised to 70 kg

Structural model:

* 1 compartment model using ODE (V, CL, ka, TLAG)
* 1st order absorption process with lag time
* 1st order elimination process

Covariate model:

* WT on CL and V following allometric principles

Variability model:

Inter-individual variability:

* + - Exponential model for V, CL, ka and TLAG (this last fix to 0.1) expressed as standard deviation
    - Correlation between CL, V and ka random variables expressed in correlation scale

Residual error model:

* Combined error model

## UseCase7

Warfarin population pharmacokinetic model using Comparments

**Dosing regimen:** single oral administration

**Dataset:**

* ID : Patient identifier (n=32)
* TIME [h]
* WT : Patient’s weight [kg]
* AMT : Total drug administered [mg]
* CMT : compartment number (1: absorption compartment)
* DVID : dependent variable identifier (0: dose,1:PK measurement)
* DV : Warfarin concentration [mg/L]
* MDV : missing dependent variable (0: observation, 1: dosing record)
* logtKG : log transformed patient’s body weight standardised to 70 kg

Structural model:

* 1 compartment model using ODE (V, CL, ka, TLAG, FORAL)
* 1st order absorption process with lag time and bioavailability (fix to 1)
* 1st order elimination process

Covariate model:

* WT on CL and V following allometric principles

Variability model:

Inter-individual variability:

* + - Exponential model for V, CL, ka and TLAG (this last fix to 0.1) expressed as standard deviation
    - Correlation between CL and V random variables expressed in correlation scale

Residual error model:

* Combined error model

## UseCase8

Warfarin population pharmacokinetic model

**Dosing regimen:** single oral administration

Dataset:

* ID : Patient identifier (n=32)
* TIME [h]
* WT : Patient’s weight [kg]
* AGE [years]
* SEX (0: female, 1: male)
* AMT : Total drug administered [mg]
* OCC : occasion identifier (1: 1st occasion , 2: 2nd occasion)
* MDV : missing dependent variable (0: observation, 1: dosing record)

Structural model:

* 1 compartment model using ODE (V, CL, ka, TLAG)
* 1st order absorption process with lag time
* 1st order elimination process

Covariate model:

* WT on CL and V following allometric principles

Variability model:

Inter-individual variability:

* + - Exponential model for V, CL, ka and TLAG (this last fix to 0.1) expressed as standard deviation

Between occasion variability:

* + - Between occasion variability on V and CL
    - Correlation between CL and V between occasion random variables expressed in correlation scale

Residual error model:

* Combined error model

## UseCase9

Warfarin population pharmacokinetic model

**Dosing regimen:** intravenous infusion

**Dataset:**

* ID : Patient identifier (n=32)
* TIME [h]
* WT : Patient’s weight [kg]
* AMT : Total drug administered [mg]
* RATE : infusion rate [mg/h]
* DV : Warfarin concentration [mg/L]
* MDV : missing dependent variable (0: observation, 1: dosing record)
* logtKG : log transformed patient’s body weight standardised to 70 kg

Structural model:

* 1 compartment model using ODE (V, CL)
* 0 order input
* 1st order elimination process

Covariate model:

* WT on CL and V following allometric principles

Variability model:

Inter-individual variability:

* + - Exponential model for V and CL expressed as standard deviation
    - Correlation between CL and V random variables expressed in correlation scale

Residual error model:

* Combined error model

## UseCase11

Poisson count data model

**Dosing regimen:** NA

Dataset:

* ID : Patient identifier (n=100)
* TIME [h]
* CP : Drug concentration acting as covariate [mg/L]
* DV : Number of counts
* MDV : missing dependent variable (0: observation, 1: dosing record)

Statistical model:

* Poisson distribution model

Covariate model:

* Linear effect of drug concentration on baseline count parameter on the logarithmic domain to ensure parameter positivity

Variability model:

Inter-individual variability:

* Exponential model for baseline count parameter expressed as variance

## UseCase14

Time to event model for exact and right censored information

**Dosing regimen:** NA

**Dataset:**

* ID : Patient identifier (n=32)
* TIME [h]
* TRT : Treatment identifier
* DV : Event identifier (0: no event or right censored, 1: event at exact time)

Statistical model:

* Constant hazard model

Covariate model:

* Proportional covariate model of treatment on the baseline hazard

## UseCase17

Warfarin population pharmacokinetic model at steady-state

**Dosing regimen:** single oral administration

**Dataset:**

* ID : Patient identifier (n=32)
* TIME [h]
* WT : Patient’s weight [kg]
* AGE [years]
* SEX (0: female, 1: male)
* AMT : Total drug administered [mg]
* SS : Steady-state
* II : Dosing interval
* DVID : Dependent variable identifier (0: dose,1:PK measurement)
* DV : Warfarin concentration [mg/L]
* MDV : Missing dependent variable (0: observation, 1: dosing record)
* logtKG : Log transformed patient’s body weight standardised to 70 kg

Structural model:

* 1 compartment model using ODE (V, CL, ka)
* 1st order absorption process
* 1st order elimination process

Covariate model:

* WT on CL and V following allometric principles

**Variability model:**

Inter-individual variability:

* + - Exponential model for V, CL and ka expressed as standard deviation
    - Correlation between CL and V random variables expressed in correlation scale

Residual error model:

* Combined error model

# Implementing M&S workflows using the ddmore R package

Along with the collection of MDL models, a set of R scripts have been prepared to illustrate how these models can be used in a M&S workflow using the “ddmore” R package:

* + - * Initialisation of the R console
      * Read and parse MDL files in R
      * Exploratory graphical analysis of the data
      * Parameter estimation with Monolix
      * Evaluation of the model executed in Monolix using Xpose
      * Parameter estimation of the same model with NONMEM
      * Evaluation of the model executed in NONMEM using Xpose
      * Change estimation method to FOCEI and re-estimation of the model via PsN
      * Bootstrap in Perl speaks NONMEM to evaluate parameter precision
      * Updating parameter estimates in the MDL Parameter Object using MLE values from NONMEM estimation
      * Performing a Visual Predictive Check (VPC) in PsN using MLE values from NONMEM
      * Simulate new observed values using the simulx function in the mlxR package.

All the R scripts have been commented to guide the tester though the code and provide information regarding the new ‘ddmore’ functions. Additional information of the functions can be obtained from the R help typing the name of the function after “?” (e.g. ? as.PharmML) or directly navigating thorough the help files.

Please note that there might be some cases were not all the steps can be performed, these information is also provided in the R script. A link to a set of html reports against which you could confront your results will be provided shortly after the release at the DDMoRe Forum (http://www.ddmore.eu/forum).

## Execution of the R script

There are different ways of executing the R scripts. We recommend to run the scripts line by line so the user gets familiarised with the different functions and they output produced. However, and given the execution of some tasks might take several minutes and in some cases up to an hour, two alternative mechanism are possible: (i) the “source” option and (ii) the “spin” function of the knitr R package.

To run the R script line by line:

Navigate to the ./scripts subfolder

Open the relevant file in the MDL-IDE editor by double-clicking. Select any code lines you wish to execute by marking them with your cursor, and press CTRL+R+R to execute them. You can also modify the code to explore different options.

The tasks has ended once the cursor in the console appears highlighted in blue again

When using this option, and depending upon the command executed, the results will be returned to your workspace (e.g. folder containing the results of a estimation in Monolix , generation of pdf files), to your console (e.g. information of the estimation process, evaluation of the results) or to the R graphics window (e.g. plot of data).

To run the R script via source:

Navigate to the ./scripts subfolder

Right click on the R file named you wish to execute. Then select --> “Run as” --> “R script in R submitting directly”.

The task has ended once the cursor in the console appears highlighted in blue again

This option will also return information as if the code would have been run line by line.

To run the R script via spin function:

load the knitr library, e.g. library(knitr)

Spin the file using the full path to the R script, e.g. spin(file.path(Sys.getenv("MDLIDE\_WORKSPACE\_HOME"),"UseCasesDemo","scripts","UC2\_Prod4.1\_Beta.R"))

The task has ended once the cursor in the console appears highlighted in blue again

This option will return the results to your workspace, and will also generate an html report collecting all the commands and their respective output.

If you have any question or experience any problem while executing the R scripts, do not hesitate to contact us via the DDMoRe Forum (<http://www.ddmore.eu/forum>).

# LIST OF KNOWN ISSUES

The following issues are known to the MDL Developers at the time of the Public Beta release of the MDL and the Interoperability Framework Standalone Execution Environment. (11th December 2015). Many of these have been noted in the description of the MDL Objects in the previous sections. They are collated here for reference.

## Data Object

### DATA\_INPUT\_VARIABLES

* If time is used as idv, the variable name used in the MDL needs to be TIME 🡪 Otherwise the SO is not correctly populated for Monolix
* For Monolix, If covariates are to be used in the model, the name given to the variable in MDL needs to match the name in the dataset header , taking also into account lower or upper cases 🡪 Monolix uses the header name to retrieve the covariate information
* Standard NONMEM ignore statements are not supported in the current version. Any data processing needs to be done before using the dataset for a task in Monolix or NONMEM. Special treatment of the header row for NONMEM is not needed since a logic to ignore that row is automatically implemented
* In NONMEM, models with dosing to multiple compartment need a CMT column in the dataset indicating to which compartment the dose is. This compartment number needs to match the ODE number specified in the DEQ section. Monolix does not support dosing to multiple compartments unless COMPARTMENTS (PK macros are used)

### SOURCE

* MDL file and data need to be collocated for Monolix execution

## Parameter Object

### STRUCTURAL

* Structural parameters without IIV associated need to be redefined in the INDIVIDUAL\_PARAMETER block, otherwise they are not correctly translated to MLXTRAN (see INDIVIDUAL\_PARAMETERS)
* Lag time parameter does not have a key word that can be recognized by the target tools, unless COMPARTMENTS (PK macros) are used to define the structural model. This means that lag time mechanism has to be explicitly defined in the model (see UseCase1 for an example) and estimation of IIV is not supported when using FOCE or FOCEI in NONMEM
* Bioavailability parameter does not have a key word that can be recognized by the target tools, unless COMPARTMENTS (PK macros) are used to define the model. This means that bioavailability has to be implemented in the model equations (see UseCase4). It also implies that bioavailability cannot be used to define initial amount in a compartment different than the one specified under the AMT column (common practice in previous versions of NONMEM to initialise compartments)

### VARIABILITY

* Correlation parameter only works in Monolix and Simulx if correlation is specified, not covariance 🡪 as a temporary solution, always use correlation scale
* When Simulx is used in a model with correlation between parameters, the default Monolix name for correlation is used (e.g. correlation between V and CL, r\_V\_CL) 🡪 this is triggered by the lack of name in MDL for the correlation parameters. To solve this the parameter name in the vector of parameters used as input in Simulx needs to map the correlation name given by Simulx.
* The specified scale (var or sd) needs to map the one indicated under the modelObj. The framework does not perform any transformation of scales and the one indicated in the modelObj is the one used in the executable model.
* The name used for the 'use is varlevel' variable in cannot contain an underscore when using NONMEM as target.

## Model Object

### COVARIATES

* Monolix makes a difference between covariates which act at the parameter level (only time invariant), and covariates that modified other variables in the model, so called regressors. MDL does not make an explicit differentiation, but covariates used to define type 3 individual parameters, or to define derived covariates that are used afterwards in this type of parameter declaration, will be translated as regressors. For the COVARIATES block to be Monolix-compatible, any covariate has to be either:
  + Used in a type 3 definition as a covariate, but not elsewhere (e.g. in MODEL\_PREDICTION)
  + Used in MODEL\_PREDICTION, but not elsewhere (e.g. in a type 3 definition)
  + Used in an equation producing a transformed covariate, but not elsewhere. It is not possible to have a chain of redefined covariates, or a redefined covariate depending on several other covariates.
* Regressors are currently not supported in Simulx
* Categorical covariates are not supported by Monolix
* Categorical covariates cannot be used in conditional statements 🡪 a walkaroun is to use the covariate as a categorical covariate instead.

### GROUP\_VARIABLES

* Any statement placed into GROUP\_VARIABLES will make the model non Monolix compatible:
  + Monolix only supports linear / “linear after transformation” relation between time invariant covariates, structural parameters and random effects (type 3 definition of individual parameters). No other transformation of the structural parameters will be compatible with Monolix.
  + Monolix does not support the declaration of constants in this block (they have to be placed under MODEL\_PREDICTION). Otherwise they will also be interpreted as parameter in the longitudinal section (ticket 359)
  + Transformation of covariates need to be placed under COVARIATES block

### INDIVIDUAL\_PARAMETERS

* To be Monolix compatible, INDIVIDUAL\_VARIABLES has to contain exclusively two types of statements:
  + type 3 declaration of individual parameters
  + Simple assignment to define parameters without IIV associated (e.g. BASE=POP\_BASE). Non-IIV parameters defined through simple assignments are supposed normally distributed, i.e. they can become negative. If positivity is desired, a type 3 definition has to be used fixing IIV to 0.
* For NONMEM support, individual parameters need to be defined in the same order as the structural and variability parameters appear in the STRUCTURAL block due to a dependency issue

### MODEL\_PREDICTION

* For the moment, variables need to be used sequentially and after they have been declared
* For Monolix compatibility, any variable used in MODEL\_PREDICTION has to be either:
  + the independent variable
  + defined in MODEL\_PREDICTION
  + declared in INDIVIDUAL\_VARIABLES in the way stated above
  + defined as a covariate in MDL, with the restriction mentioned in the covariate block (will be a regressor in PharmML),
  + defined as a data-derived variable (not fully functional yet, see ticket 435).

This implies in particular that STRUCTURAL\_PARAMETERS, VARIABILITY\_PARAMETERS, and random variables defined in RANDOM\_VARIABLES\_DEFINITION cannot be used in MODEL\_PREDICTION (an IDE validation could produce an interoperability warning at this level).

### OBSERVATION

* Only standard residual error model functions are supported. This means that equation based assignments are not supported in this block.
* For Monolix compatibility, different observation types have to use different STRUCTURAL\_PARAMETERS and random variables.
* Nonmem-formatted datasets describing time-to-event are not compatible with Monolix estimation due to the different handling of the record describing entering of the group at-risk. In Nonmem, it’s an MDV==1 entry, whereas in Monolix, it’s an MDV==0 entry.

# List of supported and unsupported features

**Interoperability framework features by release**

|  | | **Beta release** | **Future release(s)** |
| --- | --- | --- | --- |
| **High-level overview** | | | |
| Language standards | | MDL v. 7.0 | TBD |
| PharmML v. 0.6.1 |
| Key software components | | MDL-IDE v.1.3.0 | TBD |
| Monolix v. 4.3.2 |
| NONMEM v. 7.3 |
| R v. 3.0.3 |
| Xpose v. 4.5.3 |
| PsN v. 4.4.8 |
| Simulx v. 2.1.1 (mlxR) / 1.1.0 (mlxLibrary) |
| **Distribution/deployment** | | | |
| Stand-alone execution environment | | x | x |
| Integration within existing IT infrastructure | |  | x |
| **Features/tasks** | | | |
| Estimation | Monolix | x | x |
| NONMEM | x | x |
| winBUGS |  | x |
| Graphical diagnostics | R (user-defined) | x | x |
| Xpose | x | x |
| Re-sampling/simulation-based diagnostics | Bootstrap in PsN (+NONMEM) | x | x |
| VPC in PsN (+NONMEM) | x | x |
| Simulation | MATLAB |  | x |
| SimCyp |  | x |
| Simulx | x | x |
| Optimal design | PFIM |  | x |
| PopEd |  | x |
| **Models supported** | | | |
| Outcome types | Continuous (single or multiple) | x | x |
| Binary |  | x |
| Count | x[[1]](#footnote-1) | x[[2]](#footnote-2) |
| Categorical |  | x |
| Time-to-event | x[[3]](#footnote-3) | x[[4]](#footnote-4) |
| Ways to describe the model | ODEs (user-defined) | x[[5]](#footnote-5) | x5 |
| Closed form analytical solutions (user-defined) | x[[6]](#footnote-6) | x6 |
| Short-hand macro notation | x[[7]](#footnote-7) | x[[8]](#footnote-8) |
| One-line PK library functions |  | x[[9]](#footnote-9) |
| Administration/dosing/resetting support | Single or multiple dosing schedules | x | x |
| Single or multiple administration routes | x | x |
| Zero- or first-order input | x | x |
| Steady-state dosing (SS, II, ADDL) | x[[10]](#footnote-10) | x10 |
| Non-dosing related compartment resetting |  | x |
| Variability model types | Inter-individual variability | x | x |
| Inter-occasion variability | x[[11]](#footnote-11) | x |
| Higher levels of variability |  | x |
| Correlation between random effects | x[[12]](#footnote-12), [[13]](#footnote-13) | x |
| Prior distributions |  | x |
| Mixture models |  | x |
| Covariate types | Continuous (constant or time-varying) | x | x |
| Categorical | x | x |

# Future MDL plans:

## Introduction

Future versions of MDL will extend currently released functionality, scope of data types, parameter features, model and task types. We will endeavour to address limitations and workarounds in MDL for the Public Beta Release and provide a more full featured language.

Some features described below are already well advanced in definition of MDL but were not able to be implemented fully in time for the Public Beta Release. Others will need more work to fully define and specify how these features will be implemented in MDL.

It is difficult to guarantee which features will be present in future releases due to limitations of time and resource within the project. The following chapter gives the user some idea of the scope of intended features for future release.

## General features

We aim to support a process of annotation of the MDL model which will facilitate upload of annotation information to the DDMoRe model repository.

This will include annotation of units.

## Data Object

We anticipate the future versions of MDL will support:

### DATA\_INPUT\_VARIABLES block

* Extend the support for NONMEM standard features of data specification:
* Reset compartment
* Infusion rate / duration estimation
* Support for EVID
* Improved support for defining time-dependent covariates or regressors.

### SOURCE block

* Specification of a “header” row in the dataset providing data column names. This would then mean that the DATA\_INPUT\_VARIABLES block need not specify data columns in order and that any data column not appearing in DATA\_INPUT\_VARIABLES would automatically have “use is ignore”.
* Specification of a number of rows to skip before reading the data file
* Additional input data formats (beyond CSV to include e.g. SAS transport files)
* Additional data structures (beyond nonmemFormat)
* Inline data – data defined in column or R named list format.
* Data defined through an R script.

## Parameter Object

We anticipate the future versions of MDL will support:

### STRUCTURAL and VARIABILITY block

* Vector and matrix support for STRUCTURAL and VARIABILITY blocks to enable specification of multivariate distributions.
* Specification of a superset of parameters in the Parameter Object which would apply across more than one model, with the Model Object then containing component models which would have their own parameters, for example a PK component model and a PD component model.

### VARIABILITY block

* Matrix support within the VARIABILITY block to enable full specification of variance-covariance and standard deviation-correlation matrices.

## Model Object

We anticipate the future versions of MDL will support:

### RANDOM\_VARIABLE\_DEFINITION block

* Allowing the user to define individual parameter distributions (e.g. CL, V, KA), covariates distributions.
* Future versions of MDL will use the Prob-Onto ontology of random variable distributions as defined in PharmML. This will enable a wider range of distributions to be used for between subject variability, residual unexplained variability and prior distributions.

### COVARIATES block

* Interpolation in covariates / regressors through use of interpolation functions.
* Improved support for time-varying covariates / regressors to ensure that translation to target software is robust.
* Specification of covariate models including error in measurement of covariates, uncertainty in dose amounts or dose times or observation times.

### MODEL\_PREDICTION block

* Definition of the structural model using “one line” PK and PD library models.
* Mapping of standard models (specified via COMPARTMENTS and the PK library models described above) to closed-form solutions in target software (where these are available)
* The ability to combine COMPARTMENT and DEQ specification of structural models.
* The ability to specify mixture models.

### OBSERVATION block

Future versions of MDL will use the Prob-Onto ontology of random variable distributions as defined in PharmML. This will extend the range of different observation distributions which can be encoded in MDL.

#### Binary data

In this version of MDL, binary data (where outcomes are 0, 1 in the observed data) are encoded as categorical outcomes with two categories. Future versions of MDL will be able to use the Prob-Onto definitions of Bernoulli and Binomial distributions.

The syntax for Binary data outcomes is

<<OUTCOME VARIABLE NAME>> : {type is discrete

withCategories{ <<category1>>, <<category2>> },

distn = Bernoulli(category = <<category1 or category2>>,

probability = <<VARIABLE NAME>> )

}

For example (again, showing the INDIVIDUAL\_VARIABLES, MODEL\_PREDICTION and OBSERVATION blocks to show the model construction):

**INDIVIDUAL\_VARIABLES**{

logit(indiv\_BASE) = linear(**pop**= POP\_BASEP,

**ranEff**=[eta\_PPV\_EVENT],

**trans** is **logit**)

}# end INDIVIDUAL\_VARIABLES

**MODEL\_PREDICTION**{

LP = logit(indiv\_BASE) + POP\_BETA\*CP

P1 = invLogit(LP)

}# end MODEL\_PREDICTION

**OBSERVATION**{

Y : { **type** is discrete withCategories{none, event},

distn = Bernoulli(category = event,

probability = P1) }

}

Again, note that the INDIVIDUAL\_VARIABLES block defines the individual baseline by combining the population parameter and the random effect. Note also that this specification uses a logit transformation to ensure that the individual baseline indiv\_BASE variable is on the (0,1) probability scale. Then, the linear regression with plasma concentration (CP) is defined in the MODEL\_PREDICTION to facilitate interoperability across target software. Finally LP is back-transformed to the probability scale to give variable P1 which is the probability of an event to be used in the Bernoulli distribution. The Bernoulli distribution needs to have the category specified along with the associated probability. Since the model does not have an explicit link to the 0,1 outcomes in the data, we must define explicitly which category is being modelled.

#### Categorical data

In this version of MDL, categorical outcomes are modelled as a categorical distribution only – ordered categorical data or adjacent categories models must have the probability of being in each group calculated explicitly before use in the categorical distribution. Future versions of MDL will expand the definition of outcomes to other categorical outcome types using the Prob-Onto definitions.

The syntax for Categorical data outcomes is:

<OUTCOME VARIABLE NAME> : {type is categorical

withCategories{ <category1> when <VARIABLE>,

<category2> when <VARIABLE>,

…

<category N> when <VARIABLE>},

}

In the above, “<<category\_k>> when <<VARIABLE>>” the <<VARIABLE>> to be used must be on the scale (0,1) and is the probability of <<category\_k>>. This <<VARIABLE>> must be defined in the MODEL\_PREDICTION block.

For example:

**GROUP\_VARIABLES**{

B0 = Lgt0

B1 = B0 + Lgt1

B2 = B1 + Lgt2

}

**INDIVIDUAL\_VARIABLES**{

indiv\_B0 = general(**grp**=B0, **ranEff** = [eta\_PPV\_EVENT])

indiv\_B1 = general(**grp**=B1, **ranEff** = [eta\_PPV\_EVENT])

indiv\_B2 = general(**grp**=B2, **ranEff** = [eta\_PPV\_EVENT])

}# end INDIVIDUAL\_VARIABLES

**MODEL\_PREDICTION**{

EDRUG = Beta \* CP

A0 = indiv\_B0 + EDRUG

A1 = indiv\_B1 + EDRUG

A2 = indiv\_B2 + EDRUG

P0 = invLogit(A0)

P1 = invLogit(A1)

P2 = invLogit(A2)

Prob0 = P0

Prob1 = P1 - P0

Prob2 = P2 - P1

Prob3 = 1 - P2

}# end MODEL\_PREDICTION

**OBSERVATION**{

Y : {**type** is **categorical**

withCategories{ none when Prob0,

mild when Prob1,

moderate when Prob2,

severe when Prob3}

}

In the above code, the cutpoints between categories are defined in the GROUP\_VARIABLES block (B0, B1, B2) and individual values for these are defined in the INDIVIDUAL\_VARIABLES block. The linear effect of CP (plasma concentration) is defined in the MODEL\_PREDICTION block and this is added to the individualised cutpoints (A0, A1, A2). These are then back-transformed to the probability scale (P0, P1, P2) and the ordered categorical model is defined by calculating the probability of each category as the difference from the previous category – Prob0, Prob1, Prob2, Prob3.

#### TIME TO EVENT

Time to event definition in MDL will be extended to cover interval censoring and repeated time to event.

## Task Properties Object

We anticipate the future versions of MDL will support:

* The ability to define target specific settings for algorithms.
* Define a Simulation task
* Define an Optimal design task

## MOG Object

We anticipate the future versions of MDL will support:

* The ability to define variable name mapping between MDL Objects e.g. define how a variable name defined in the Data Object maps to a Model Object variable. This will further enable the modularity and independence of MDL Objects.

## Design Object

The Design object will describe the trial design for use in simulation or design evaluation. It may alternatively include elementary trial designs and constraints for use in finding optimal designs.

## Prior Object

The Prior object will be used to describe prior distributions on model parameters for use in Bayesian tasks such as estimation. It will support parametric, non-parametric and empirical distributions for parameters.

# Glossary

## Acronyms and Abbreviations

|  |  |
| --- | --- |
| Acronym | Definition |
| AMT | Dose amount |
| BOV | Between Occasion Variability (synonym for IOV - Inter-occasion variability) |
| COV | Covariance |
| CORR | Correlation |
| CTS | Clinical Trial Simulation |
| CWRES | Conditional Weighted Residual |
| DDMoRe | Drug Disease Model Resources |
| DoW | Description of Work |
| DV | Dependent Variable |
| EFPIA | European Federation of Pharmaceutical Industries and Associations |
| EMA | European Medicines Agency |
| EPS | Epsilon - Residual Unexplained Variability random effect |
| ETA | Empirical Bayes prediction of the inter-individual random effect in a PK or PD parameter |
| FDA | Food and Drug Administration |
| FIS | Framework Integration Service |
| ID | Individual |
| IDV | Independent variable |
| II | Inter-dose Interval |
| IMI | Innovative Medicines Initiative |
| IMI-JU | Innovative Medicines Initiative Joint Undertaking |
| IOF | Interoperability framework |
| IPRED | Individual Prediction |
| IWRES | Individual Weighted Residual |
| MDL | Model Description Language |
| MDL-IDE | Modelling Definition Language Integrated Development Environment |
| MDV | Missing Dependent Variable |
| MIF | Mango Integration Framework |
| MOG | Modelling Object Group |
| NONMEM | NONLinear Mixed Effects Modelling (see Names) |
| OBS | Observed (value or data) |
| PharmML | Pharmacometrics Markup Language |
| PD | Pharmacodynamic |
| PK | Pharmacokinetic |
| PK/PD | Pharmacokinetic - Pharmacodynamic modelling |
| PRED | Population Prediction |
| PROV-O | Provenance Ontology |
| PsN | Perl Speaks NONMEM  <http://www.uppsala-pharmacometrics.com/software.html> |
| RDF | Resource Description Framework |
| RUV | Residual Unexplained Variability |
| sd | Standard Deviation |
| SE | Standard Error |
| SEE | Stand-alone Execution Environment |
| SO | Standard Output |
| TEL | Task Execution Language.  The working name within the DDMoRe project for the ddmore R package used to perform tasks with MDL and define pharmacometric workflow. |
| TES | Task Execution Server |
| var | Variance |
| VPC | Visual Predictive Check |
| WP | Work Package |
| WRES | Weighted Residual |
| XML | Extensible Markup Language |

## Definitions and System Names

|  |  |
| --- | --- |
| System Name | Description |
| Annotation | A description attached to a model or element of a model; see RDF triple. |
| Bootstrap | Bootstrap is a tool for calculating bias, standard errors and confidence intervals of parameter estimates. It does so by generating a set of new datasets by sampling individuals with replacement from the original dataset, and fitting the model to each new dataset |
| Connector | A piece of software which enables modelling software to communicate with the interoperability framework |
| Converter | A piece of software which enables translation across languages (e.g. mdl to pharmML |
| End-User | A specific user role with privileges to manage only his/her own jobs queues, etc. |
| Extensible Markup Language (XML) | An open standard for exchanging structured documents and data over the internet that was introduced by the World Wide Web Consortium (W3C). |
| Metadata | Metadata (metacontent) is defined as data providing information about one or more aspects of the data, such as:   * Means of creation of the data * Purpose of the data * Time and date of creation * Creator or author of data * Placement on a [computer network](http://network) where the data was created * [Standards](http://standard) used |
| Monolix | A software for the analysis of nonlinear mixed effects models |
| MDL-IDE | Graphical user interface of the Interoperability Framework. It provides a framework within which files containing MDL code can be created and edited and Modelling & Simulation workflows can be created and executed. |
| MDL | MDL is the Model Description Language (formerly MCL - Model Coding Language) the human writable and human readable language designed to describe pharmacometric models. |
| MLXTRAN | The language used to define models that are executed with Monolix. |
| NM-TRAN | The language used to define models that are executed with NONMEM. |
| NONMEM | A software for the analysis of nonlinear mixed effects models.  <http://www.globomax.com/nonmem.htm> |
| [Ontology](http://www.globomax.com/nonmem.htm) | [An organization of some knowledge domain that is hierarchical and contains all the relevant entities and their relations.](http://www.globomax.com/nonmem.htm)  An ontology is used to define the relationships and objects that are used to define the RDF Triples that describe the data, models and results with a Pharmacometrics Workflow |
| R | R is a free software environment for statistical computing and graphics.  <https://www.r-project.org/> |
| RDF Triple | An RDF Triple is a statement which relates one object to another. It is composed of three parts:   * the subject - the entity we are describing * a predication - the relationship * the object - the description   DDMoRe uses RDF triples to describe models held within the repository, for example:  “MODEL-000034765” “has author” “Lena Friberg” |
| Task Execution Service (TES) | Performs job-management within the Interoperability Framework. |
| WinBUGS | Windows implementation of the BUGS (Bayesian Inference Using Gibbs Sampling) project, concerned with flexible software for the Bayesian analysis of complex statistical models using Markov chain Monte Carlo (MCMC) methods.  <http://www.mrc-bsu.cam.ac.uk/bugs/winbugs/contents.shtml> |
| [Pharmacometric Workflow](http://www.mrc-bsu.cam.ac.uk/bugs/winbugs/contents.shtml) | [Tracking the evolution of a model and associated inferences from initial model to final model, capturing metadata and annotations that will facilitate creation of a run record, audit log, QC and reproducibility of all steps within the workflow. Each step in the Pharmacometric workflow may consist of a Task Workflow which defines the procedural steps required to perform a sequence of tasks for a given model.](http://www.mrc-bsu.cam.ac.uk/bugs/winbugs/contents.shtml) |
| Task Workflow | A sequence of tasks and procedural steps which can be captured in a scriptable language, like R, facilitating reproducibility of the outputs for a given set of inputs. |
| Interoperability | One stated of the DDMoRe project is to provide the capability to define the model once and then use it across a variety of target software tools. We call this “Interoperability”. |
| Repository | Another stated aim of the DDMoRe project is to provide a library platform for pre-competitive sharing of models - disease models, drug models etc. We call this library the “Repository”. |
| PharmML | XML based exchange format for encoding of non-linear mixed effect models, trial design and modelling steps used in pharmacometrics. URL: pharmml.org |
| Standard Output (SO) | Tool-independent exchange format intended for storage of results in standardised form, enabling effective data exchange within complex workflows as well as to support the user in assessing, reviewing and reporting a modelling step. |

1. Poisson-type. [↑](#footnote-ref-1)
2. Non-Poisson type. [↑](#footnote-ref-2)
3. Exact/right-censored type. NOTE: It is currently not possible to combine a PK model directly with a time-to-event PD model. PK concentrations should instead be included as a column in the data set. [↑](#footnote-ref-3)
4. Incl. interval-censored type. [↑](#footnote-ref-4)
5. Resolves to: ADVAN 13 + $DES in NONMEM; ODEs in Monolix. [↑](#footnote-ref-5)
6. Resolves to: $PRED in NONMEM; analytical solution in Monolix. [↑](#footnote-ref-6)
7. Resolves to: ADVAN 13 + $DES in NONMEM; PK macros in Monolix. [↑](#footnote-ref-7)
8. Resolves to: ADVAN 1-4/10-12 in NONMEM if one of these types, otherwise to ADVAN 13 + $DES; PK macros in Monolix. [↑](#footnote-ref-8)
9. Resolves to: ADVAN 1-4/10-12 in NONMEM if one of these types, otherwise to ADVAN 13 + $DES; PK macros in Monolix. [↑](#footnote-ref-9)
10. There is a known bug in the current version of Monolix, which means that ADDL is not handled properly. [↑](#footnote-ref-10)
11. Inter-occasion variability is not currently supported in the translation to Monolix. [↑](#footnote-ref-11)
12. There is a limitation in the current translation to NONMEM, which means that random effects parameters for inter-individual as well as inter-occasion variability have to be defined in MDL in an order that ensures that the resulting OMEGA matrix in NONMEM is a block-diagonal band matrix. [↑](#footnote-ref-12)
13. There is a known bug in the current translation to NONMEM, which means that the OMEGA BLOCK statement for correlated parameters will always appear before the OMEGA statements for un-correlated parameters. To ensure a correct model in NONMEM, the user must make sure to define the correlated random effects parameters before the uncorrelated random effects parameters in MDL. [↑](#footnote-ref-13)