PK macros translation rules (with examples)

Maciej Swat (EMBL-EBI)
16 April 2015

Assumptions

All translation rules are based on the following assumptions:

1. the PK macros to be translated are correctly encoded - this means that no validation rules are formulated and enforced (for now) and that the validation should be defined as a separate task
2. no defaults are used, i.e.
    a. macros use always the '**cmt**' argument, where applicable
    b. (a) means that following macro is valid: **iv()** which corresponds to **iv(cmt=1)** - it is when only one compartment is defined and therefore the 'cmt' argument is not required
    c. '**adm/type**' is always required even if there is a single route of administration
    d. '**amount**' argument in the <peripheral> macro is required

Proposed strategy

1. Start with real-life examples, so called NONMEM PREDPP library models: ADVAN1-4 & 10-12 (see example section below). This is a set of well-known and frequently used models in pharmacometrics (described in detail in the PK macro report [2]), then go to more complex examples.
2. Models mentioned in (1) must be correctly formulated as full macros, i.e. no defaults.
3. When testing more general macros, making use of defaults, one should
    a. first, complete the given set of macros with missing arguments, e.g. instead of *'iv()'* write '*iv(cmt=1)*' etc.
    b. process then the complete macros using the translation rules.

Translation rules

This is a set of rules collected while doing the R converter for the PK macros [3]. See [2] for the full set of macros and arguments. The original document [1] from Lixoft describes the elements and their meaning in detail.

1. Start with (1) *'compartment'* and (2) *'peripheral'* macros in that order - they define the core structure - followed by remaining ones

2. Identify associated
    a. compartment numbers, e.g. *'cmt=1'* - store in *cmtNumber* array
    b. amounts names, e.g. *'amount=Ac'* - store in *cmtAmount* array
    c. volume names, e.g. *'volume=V'* - store in *cmtVolume* array
    d. concentration names, e.g. *'concentration=C'* - store in *cmtConcentration* array

3. For each '**compartment**' macro, e.g.
    ```
    compartment(cmt=1, amount=Ac, volume=V, concentration=C)
    ```
    create an empty ODE 'd*cmtAmount*[cmt]/dt = ', i.e. 'dAc/dt = '

4. For each '**peripheral**' macro, e.g.
    ```
    peripheral(k12, k21, amount=Ap)
    ```

a. create an empty ‘j’ ODE ‘d*cmtAmount*[j]/dt = ’, i.e. ‘dAp/dt = ’
b. process ‘kij’ or ‘k_i_j’arguments - label ‘i’ is for one of the central ‘compartment’s defined before, label ‘j’ is for the current peripheral compartment
    i. Add ‘- kij * *cmtAmount*[j]’ to the ‘i’ ODE
    ii. Add ‘+ kij * *cmtAmount*[i]’ to the ‘current’ ODE
c. process ‘kji’ or ‘k_j_i’arguments
    i. Add ‘- kji * *cmtAmount*[i]’ to the ‘j’ ODE
    ii. Add ‘+ kji * *cmtAmount*[j]’ to the ‘current’ ODE

5. ‘**absorption/oral**’ macro - three options: {either ‘Tk0’, ‘ka’ or ‘ka, Ktr,Mtt’}

```
oral(cmt=i, type/adm, Tlag, p, ka, Ktr/Mtt)
```

or

```
absorption(cmt=i, type/adm, Tlag, p, Tk0)
```

a. each such macro means a new ‘depot’ compartment
b. create new ODE string ‘d*cmtAmount*[new name]/dt = ’, i.e. ‘dAd/dt = ’
c. identify the target compartment, ‘i’
d. update compartment/amount ‘arrays’
e. case1: *zero order* absorption, example 9

```
absorption/oral(adm=j, cmt=i, Tk0)
```

    i. add ‘- Tk0’ to the new ODE string
    ii. add ‘+ Tk0’ to the ‘i’ target compartment ODE
f. case2: *first order* absorption must have ‘**ka**’ argument

```
oral(adm=a, cmt=i, ka)
```

    i. add ‘+ ka**cmtAmount*[new ‘j’]’ to the target compartment ‘i’
    ii. add ‘- ka**cmtAmount[*new ‘j’]’ to the current depot compartment ODE
g. case3: (example 11) models with *transit* compartments absorption have additional ‘**Ktr**’ and ‘**Mtt**’

```
oral(adm=a, cmt=i, ka, Ktr, Mtt)
```

    i. new absorption compartment, e.g. ‘Aa’
    ii. add ‘+ ka*Aa’ to the target compartment ‘i’
    iii. add the following new ODE *‘dAa/dt = exp[log(F*Dose)) + log(Ktr) + n*log(Ktr*(t-t_Dose)) - Ktr*(t-t_Dose) - log(n!)] - ka*Aa’*
    iv. for simplicity (ii) assumes there is only one administration defined with ‘Ktr/Mtt’, otherwise need more As’s compartments and related ODs, i.e. Aa1 with ‘dAa1/dt=...’
    v. target ‘*Dose*’
h. As the last step one needs to provide the information about the input, required for the link with a dataset
    i. the input is ‘oral’
    ii. administration number is given by ‘adm/type’
    iii. target *cmtAmount*[i] (except case 3)
    iv. i-iii: New *Input[inputNumber] ORAL administration, adm=a, target=cmtAmount[i]*

6. ‘**iv**’ macro

```
iv(cmt=i, type/adm=a, Tlag , p)
```

a. in this case there are NO additions to ODEs
b. only the input information needs to be provided, similar to the ‘oral’ case
    i. the input is ‘iv’
    ii. administration number is given by ‘adm/type’
    iii. target *cmtAmount*[i]
    iv. i-iii: *Input[inputNumber] IV administration, adm=a, target=cmtAmount[i]*

7. '**transfer**' macro

```
transfer(from=i, to=j, kt=kl)
```

   a. note here an assignment for kt, 'kl' to be used
   b. extract '**to**' - target compartment number, '**from**' - source compartment number
   c. assign to 'to' compartment ' + **kl****cmtAmount*[i]'
   d. assign to 'from' compartment ' - **kl****cmtAmount*[j]'

8. '**elimination**' macro - three options {either 'k', 'Km&Vm' or 'CL&V'}
   a. case 1: '**k**' linear elimination

```
elimination(cmt=i, k)
```

   add to the 'cmt' compartment the ' - k**cmtAmount*[i]'
   b. case 2: linear elimination with CL

```
elimination(cmt=i, volume=V, CL)
```

   add to the 'cmt' compartment the ' - CL/V**cmtAmount*[i]'
   c. case3: '**Km & Vm**' saturable elimination

```
elimination(cmt=i, Km, Vm)
```

   add to the 'cmt' compartment the ' - Vm**cmtAmount*[i]/(Km + *cmtAmount*[i])'

9. '**effect**' macro - see example 8

```
effect(cmt=i, ke0, concentration=Ce)
```

   a. update compartment/volume/amount 'arrays' although the last two will have 'NaN's for this macro
   b. create new algebraic equation *cmtConcentration*[i] = *cmtAmount*[i]/*cmtVolume*[i], i.e. C= Ac/V
   c. create a new ODE 'dCe/dt = ke0*(*cmtConcentration*[i] - Ce)'

10. '**depot**' macro - is a bit special - it occurs only in connection with explicitely defined ODE's in the MLXTRAN literature, e.g.

***PK:** depot(adm=a, target=Ac)*
***EQUATION:** ddt_Ac = -k*Ac*
and means bolus IV administration
or
***PK:** depot(adm=a, target=Ac, ka)*
***EQUATION:** ddt_Ac = -k*Ac*
and means ORAL administration

   a. case 1: without '**ka**' argument

```
depot(adm=a, target=Ac)
```

   no additions to the ODEs - only to '*Input[inputNumber]*' for a new *'inputNumber'*
      i. New *Input[inputNumber] IV administration, adm=a, target=cmtAmount[i]*
   b. case 2: with '**ka**' argument

```
depot(adm=a, target=Ac, ka)
```

   corresponds to these 2 macros

```
compartment(cmt=1, amount=Ac)
oral(cmt=1, ka)
```

      i. i.e. creates new *depot* compartment and according ODE 'd*cmtAmount*[new depot name]/dt = ', i.e. 'dAd/dt = - ka*Ad'
      ii. adds ' + ka*Ad' to the `target` compartment, `Ac`.
      iii. NEW *Input[inputNumber] ORAL administration, adm=a, target=cmtAmount[i]*

Examples

The following 'ODEs' and 'Input's have been generated by the set of R scripts - see [3] - NOTE, the scripts underwent a number of changes - to be updated soon.

Example 1: ADVAN1, TRANS1

Macros:
```
compartment(cmt=1, amount=Ac, volume=V)
iv(adm=1, cmt=1)
elimination(cmt=1, k)
```

ODEs:
```
dAc/dt= - k*Ac
```

Input:
```
Input[1]: IV administration, adm=1, target=Ac
```

Example 2: ADVAN2, TRANS1

Macros:
```
compartment(cmt=1, amount=Ac, volume=V)
oral(adm=1, cmt=1, ka)
elimination(cmt=1, k)
```

ODEs:
```
dAc/dt= + ka*Ad2 - k*Ac
dAd2/dt= - ka*Ad2
```

Input:
```
Input[1]: ORAL administration, adm=1, target=Ad2
```

Example 3: ADVAN3, TRANS1

Macros:
```
compartment(cmt=1, amount=Ac, volume=V)
peripheral(k12, k21, amount=Ap)
iv(adm=1, cmt=1)
elimination(cmt=1, k)
```

ODEs:
```
dAc/dt= - k12*Ac + k21*Ap - k*Ac
dAp/dt= k12*Ac - k21*Ap
```

Input:
```
Input[1]: IV administration, adm=1, target=Ac
```

Example 4: ADVAN4, TRANS1

Macros:
```
compartment(cmt=1, amount=Ac, volume=V)
peripheral(k12, k21, amount=Ap)
oral(adm=1, cmt=1, ka)
elimination(cmt=1, k)
```

ODEs:
```
dAc/dt= - k12*Ac + k21*Ap + ka*Ad3 - k*Ac
dAp/dt= k12*Ac - k21*Ap
dAd3/dt= - ka*Ad3
```

Input:
```
Input[1]: ORAL administration, adm=1, target=Ad3
```

Example 5: ADVAN10, TRANS1

Macros:
```
compartment(cmt=1, amount=Ac, volume=V)
```

```
iv(adm=1, cmt=1)
elimination(cmt=1, Km, Vm)
```

ODEs:
```
dAc/dt= - Vm*Ac/(Km + Ac)
```

Input:
```
Input[1]: IV administration, adm=1, target=Ac
```

Example 6: ADVAN11, TRANS1

Macros:
```
compartment(cmt=1, amount=Ac, volume=V)
peripheral(k12, k21, amount=Ap1)
peripheral(k13, k31, amount=Ap2)
iv(adm=1, cmt=1)
elimination(cmt=1, k)
```

ODEs:
```
dAc/dt= - k12*Ac + k21*Ap1 - k13*Ac + k31*Ap2 - k*Ac
dAp1/dt= k12*Ac - k21*Ap1
dAp2/dt= k13*Ac - k31*Ap2
```

Input:
```
Input[1]: IV administration, adm=1, target=Ac
```

Example 7: ADVAN12, TRANS1

Macros:
```
compartment(cmt=1, amount=Ac, volume=V)
peripheral(k12, k21, amount=Ap1)
peripheral(k13, k31, amount=Ap2)
oral(adm=1, cmt=1, ka)
elimination(cmt=1, k)
```

ODEs:
```
dAc/dt= - k12*Ac + k21*Ap1 - k13*Ac + k31*Ap2 + ka*Ad4 - k*Ac
dAp1/dt= k12*Ac - k21*Ap1
dAp2/dt= k13*Ac - k31*Ap2
dAd4/dt= - ka*Ad4
```

Input:
```
Input[1]: ORAL administration, adm=1, target=Ad4
```

Example 8: Model with effect compartment

Macros:
```
compartment(cmt=1, amount=Ac, volume=V, concentration=C)
iv(adm=1, cmt=1)
elimination(cmt=1, k)
effect(cmt=1, ke0, concentration=Ce)
```

ODEs:
```
dAc/dt= - k*Ac
dCe/dt= ke0*(C - Ce)
C=Ac/V
```

Input:
```
Input[1]: IV administration, adm=1, target=Ac
```

Example 9: Model with oral and Tk0

Macros:
```
compartment(cmt=1, amount=Ac, concentration=Cc, volume=V)
oral(adm=1, cmt=1, Tk0)
elimination(cmt=1, k)
```

ODEs:
```
dAc/dt= + Tk01 - k*Ac
dAd2/dt= - Tk01
```

Input:
```
Input[1]: ORAL administration, adm=1, target=Ad2
```

Example 10: *Sequential zero order/first order absorption processes - only one 'adm'*

Macros:
```
compartment(cmt=1, amount=Ac, concentration=Cc, volume=V)
oral(adm=1, cmt=1, Tk0, p=F0)
oral(adm=2, cmt=1, ka , Tlag=Tk0 , p=1-F0)
elimination(cmt=1, k)
```

ODEs:
```
dAc/dt= + Tk01 + ka*Ad3 - k*Ac
dAd2/dt= - Tk01
dAd3/dt= - ka*Ad3
```

Input:
```
Input[1]: ORAL administration, adm=1, target=Ad1; p=F0
Input[2]: ORAL administration, adm=2, target=Ad3; Tlag=Tk0; p=1-F0
```

Example 11: Model with *transit* compartments. 'example_1comp_kaKtrMtt_k.txt'

Macros:
```
compartment(cmt=1,amount=Ac,volume=V,concentration=C)
oral(adm=1, cmt=1, Mtt, Ktr, ka)
elimination(cmt=1, k)
```

ODEs:
```
dAc/dt= + ka*Aa - k*Ac
dAa/dt = exp[log(F*Dose)) + log(Ktr) + n*log(Ktr*(t-t_Dose)) - Ktr*(t-t_Dose)
- log(n!)] - ka*Aa
```

Input:
```
Input[1]: ORAL administration, adm=1, target=Dose
```

References

[1] MLXTRAN_forMonolix_May2014.pdf - copy in the folder ‘PK macros’
[2] PKmacros_in_PharmML0.6_17Feb2015.pdf - copy in the folder ‘PK macros’
[3] PKmacros2ODEs.zip - R converter - copy in the folder ‘PK macros’
- run the ‘testRun.R’ script, the macro sets for ADVAN models are defined in line 10.