PhUSE 2009

Paper AD03

Web-Based Application for the Reporting of Population PKPD Models

Richard Pugh, Mango Solutions, Chippenham, UK
Justin Wilkins, Novartis, Basel, Switzerland

ABSTRACT

Pharmacometrics plays a key role in drug development, allows us to understand the manner in which a drug moves through the body, and how the resulting concentration at the target site affects measures of interest. NONMEM is the gold standard software for fitting nonlinear mixed-effects PK/PD models. Developed in the 1970s, NONMEM is a Fortran-based system that returns its analysis results as dense, hard-to-read ASCII text. A typical study may include hundreds of NONMEM model fits; understanding, comparing and reporting these runs quickly and effectively helps support efficient decisionmaking with respect to drug efficacy and safety..

NONMEM

A NONMEM model is typically specified using a “control” or “model” file; this is a text file that describes the input data, the model to be fit, the starting values for model parameters and the required outputs. A simple (!) example of a control file is shown here. Note that elements of a control file are indicated with keywords prefixed by dollar symbols (eg. $DATA specifies the input dataset).

```
$PROB Simple Run Example
```

Run Title

```
$DATA data5 IGNORE=@
$INPUT SID SEX AGE RACE HT SMOK HCTZ PROP CON AMT WT TIME SECR
       DV DROP=RATE EVID SS II ID OCC
```

Input Dataset & variables contained

```
$SUBROUTINE ADVAN2 TRANS2
$PK
    TVCL=THETA(1)
    TVV=THETA(2)
    TVKA=THETA(3)
    CL=TVCL*EXP(ETA(1))
    V =TVV *EXP(ETA(2))
    KA=TVKA*EXP(ETA(3))
    S2=V

$ERROR
    IPRED = F
    IRES  = DV - F
    W     = F
    IF(W.EQ.0) W = 1
    IWRES = IRES/W
    Y     = IPRED + W*EPS(1)
```

Model Definition

```
$THETA 18.7 87.3 2.13
$OMEGA .128 .142
$OMEGA 1.82
$SIGMA  0.0231
```

Initial Model Parameter Values

```
$TABLE ID  TIME IPRED IWRES NOPRINT ONEHEADER FILE=sdtab1
$TABLE ID  CL V KA NOPRINT  ONEHEADER FILE=patab1
$TABLE ID AGE HT WT SECR NOPRINT  ONEHEADER FILE=cotab1
$TABLE ID SEX RACE SMOK HCTZ PROP CON NOPRINT  ONEHEADER FILE=catab1
$TABLE ID  OCC TIME IPRED IWRES NOPRINT ONEHEADER FILE=mutab1
$TABLE SID NOPRINT ONEHEADER FILE=mytab1
```

Required (Data) Outputs

This definition, together with an input data (specified as "data5" in the above file) is fed into NONMEM. NONMEM then executes the run, which may take as long as several weeks to complete (this example is closer to a few seconds). When NONMEM completes, a number of textual outputs are produced, including:

- Output data files specified by "$TABLE" files in the input (6 in this file)
- A textual output file containing estimates and matrices
- Convergence information
- Various log files

A (simplified) sample from the output file, containing some parameter estimates, is shown here:

```
 ****************************************
 **                                    **
 **         FINAL PARAMETER ESTIMATE   **
 **                                    **
 ****************************************
```

```
THETA - VECTOR OF FIXED EFFECTS PARAMETERS   *********

            TH 1      TH 2      TH 3

         1.96E+01  8.46E+01  1.66E+00
```

Final Estimates of Theta

```
OMEGA - COV MATRIX FOR RANDOM EFFECTS - ETAS  ********

            ETA1      ETA2      ETA3

 ETA1
+         1.64E-01

 ETA2
+         0.00E+00  1.65E-01

 ETA3
+         0.00E+00  0.00E+00  1.30E+00
```

Final Estimates of ETA

```
SIGMA - COV MATRIX FOR RANDOM EFFECTS - EPSILONS  ****

            EPS1

 EPS1
+         2.02E-02
```

Final Estimates of Epsilon

A study may contain hundreds of NONMEM runs, all producing outputs such as those shown above. Additionally, the role of simulation has become more and more important in the model selection process, resulting in the need to produce even greater volumes of data. In all cases, model evaluation requires the extraction and comparison of key elements from the output files, as well as graphical methods for data outputs, and the ability to export these results to written reports in a reproducible manner.

CURRENT PRACTICE

Novartis' modeling and simulation (M&S) group contains 70+ members, split across geographic locations. The primary M&S locations are Basel (Switzerland), East Hanover (New Jersey) and Cambridge (Massachusetts). Many associates will execute NONMEM models each week and, like most M&S groups globally, there are many different ways of model reporting. Some of the common methods for reporting models are shown here:

Task	Possible Solution
Extracting information from output files	SAS (ad hoc, user-written code or downloaded code) S-PLUS (ad hoc, user-written code or downloaded code) R (ad hoc, user-written code, downloaded code, available libraries: Xpose, MIfuns, wfn, RNONMEM, …) MATLAB (ad hoc, user-written code or downloaded code) Census Perl or another text formatting language Manual task
Creating graphics from output data	SAS (ad hoc, user-written code or downloaded code) S-PLUS (ad hoc, user-written code or downloaded code) R (ad hoc, user-written code, downloaded code, available libraries: Xpose, RNONMEM, …) MATLAB (ad hoc, user-written code or downloaded code) Census Excel (via manual data input) Using NONMEM's "standard" Graphics
Creating comparison reports	SAS (ad hoc, user-written code or downloaded code) S-PLUS (ad hoc, user-written code or downloaded code) R (ad hoc, user-written code, downloaded code, available libraries: Xpose, RNONMEM, …) MATLAB (ad hoc, user-written code or downloaded code) Census Excel (via manual data input) Manual Task

Creating output documents recording the key results is typically a manual (or at most semi-automated) task.

THE PROJECT

OVERVIEW AND AIMS

Novartis instigated a long term project to provide model comparison and reporting capabilities. The key aims of this project were:

- To standardize the manner in which reporting was performed at Novartis M&S
- To automate the generation of reporting documents with a complete audit trail for reproducibility
- To provide non-programmers with all available functionality with the need to learn specific technologies
- To implement a flexible reporting platform that could be extended to meet future needs

INITIALISATION

The first part of this project focused on the reporting of population PKPD models based on NONMEM only. Further development phases aim to generalize the capabilities of this first phase to provide a reporting platform for other model types and modeling technologies.

Novartis invited a number of vendors to take part in an RFP process during the summer of 2008. Mango Solutions were selected as the vendor for this project in October 2008, with the project start in November 2008.

PROJECT TEAM

Novartis formed a project team consisting of business users, analysts and IT personnel in order to drive the project requirements. The development of the software was run from the UK offices of Mango Solutions. The complete project team for the PopPKPD Reporting project were as follows:

Company	Role	Name
Novartis	Project Manager	Paul Lewandoski
Novartis	IT Expert	Peter Forster
Novartis	Business Leads	Justin Wilkins, Celine Dartois Sarr
Novartis	Testers	Too many to mention!
Mango	Project Manager	Dylan Browne
Mango	Developer	Fan Shao
Mango	Developer	Feng Zhu
Mango	Developer	Geoff Gibbs
Mango	R Consultant	Francisco Gochez
Mango	Business Consultant	Richard Pugh

REQUIREMENTS

The high level project requirements can be summarized as follows:

- Provide a distributed system that can support minimally 30 concurrent users
- Users should be able to automatically import and store NONMEM run information
- Users should be able to compare runs in a tabular or graphical fashion
- Users should be able to evaluate individual runs in a tabular or graphical fashion
- Users should be able to produce standard tabular and graphical outputs based on the stored information (a list of 86 items was provided)
- Users should be able to able to control various aspects of the graphics and tables produced
- Users should be able to design and produce output reports that are tracked against the data for audit purposes
- Standard templates should be available to be run in order to generate “standard” reports
- The system should be extendable by someone able to code in the graph/table technology
- The system should reflect the Novartis security system
- All system activities should be logged as an audit trail

DEVELOPMENT PROCESS

The system requirements were fairly well understood up front, suggesting the use of the more traditional waterfall model for development. However, an iterative approach was selected in order to ensure regular user feedback to guide the development of the system. This “semi-Agile” approach worked well, particularly in the areas of system usability, where users were able to quickly suggest improvements during the development cycle. The only challenges found using this approach were the balancing of the user requests against the fixed deadline, but a positive working attitude ensured that any challenges were quickly overcome.

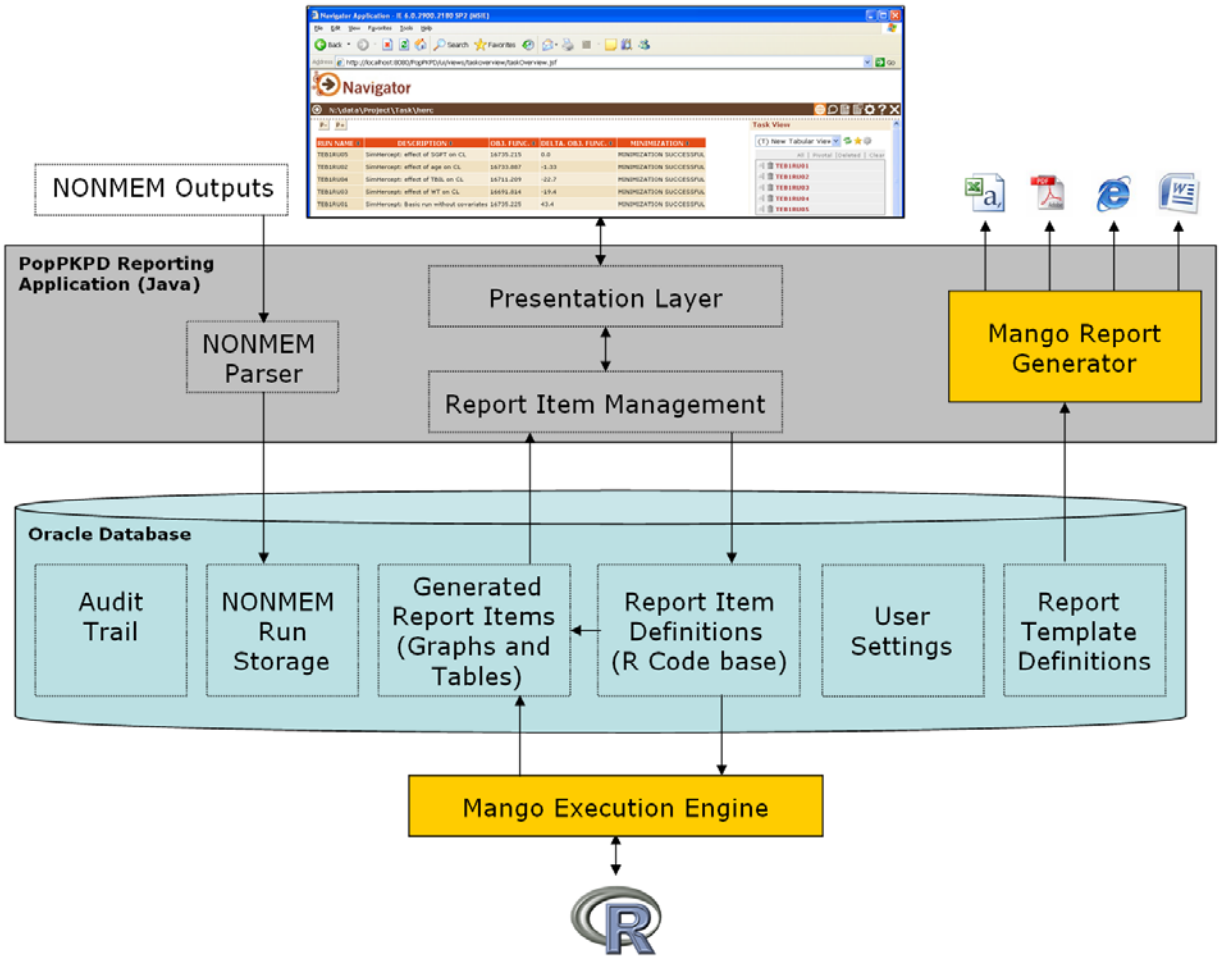
TECHNICAL SOLUTION

OVERVIEW

The designed system consisted of the following primary components:

- An Oracle database
- A Java application deployed using Apache Tomcat
- An installation of R serving the graphics and tables
- 2 of Mango’s primary (Java) development components (the Mango Execution Engine and the Mango Report Generator)

A system diagram is shown here:



ORACLE DATABASE

The database is used for a number of purposes within the PopPKPD Reporting Application. The primary uses of the database are as follows:

- NONMEM Run Information – NONMEM run information is imported and stored in a NONMEM-agnostic format within the database
- Audit Trail – Auditing information is logged and stored in the database in order to help resolve issues and to support GxP compliance
- Generated Report Items – All generated graphics and tables are stored in a compressed form within the database, along with the plot data reference and code used
- Report Item Definitions – All possible report items are stored within the database and reflected via the user information. Each definition includes information such as:
    - The type of item it will produce (eg. table or graph)
    - The (R) code
    - A set of user options
    - Versioning information
    - A category in order to organize the display
    - The data columns required in order to run the report item
- User Settings – User preferences pertaining to the report items shown, the login details, style preferences, favourite views etc.
- Report Template Definitions – Instances of “report item definitions” together with a report layout that, together, form a report template that can be viewed and exported to external formats using the Mango Report Generator

Note: Here we specify an Oracle database. However, the management of the information is governed by the Java Hibernate technology, allowing other databases to be used. For example, a MySql database was used for the early development testing.

MANGO EXECUTION ENGINE

The Mango Execution Engine (MEE) is a Java component that allows the execution of external systems in a managed way. The variety of systems that can be called from the MEE includes SAS, NONMEM, BUGS, R and S-PLUS. In this system we use the MEE in order to execute tasks within R.

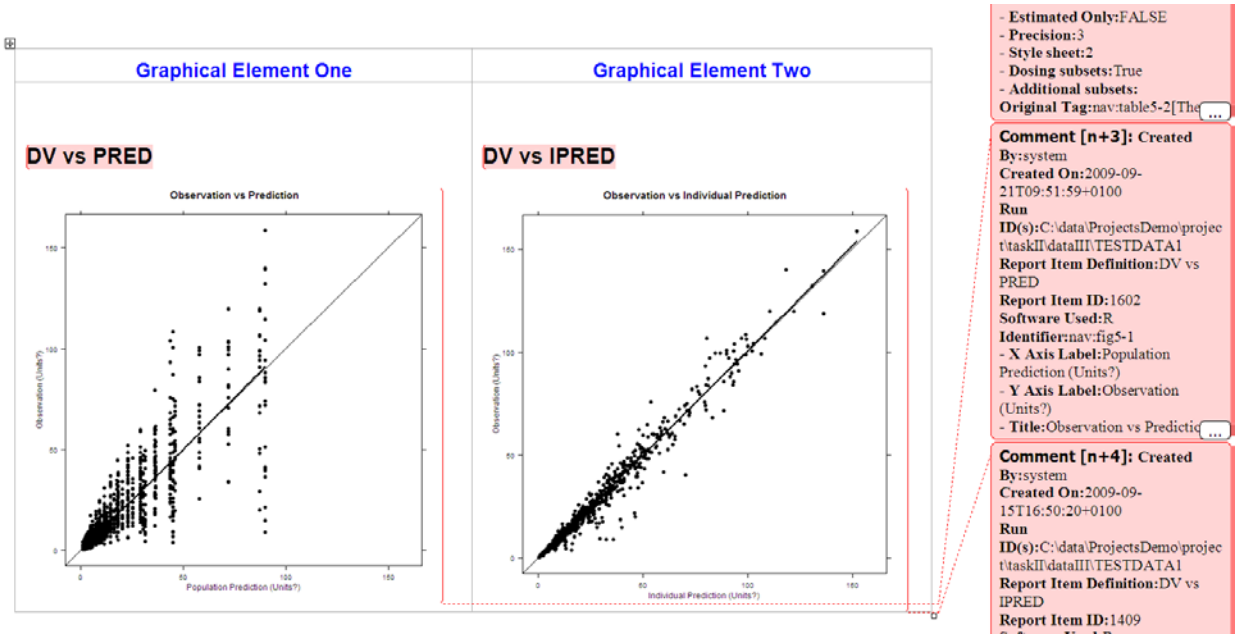
Currently this is executed on the same server as the application, but a future enhancement is to move this to a remote server; a benefit of the MEE is that no application code needs to be altered which this change occurs.

The MEE allows R to run, report errors, report warnings and return outputs of type “data”, “graph”, “table”, “text” and “error”. These are understood by the MEE, checked to ensure the correct structure is presented, then returned to the application along with the original calling information. The returns from R are stored in the database and passed to the application for presenting in the user interface.

MANGO REPORT GENERATOR

The Mango Report Generator (MRG) is a Java component that allows users to combine graphics and tables (either already generator, or to be generated in the future) into a defined template based on a specified layout. The MRG is primarily used to write these reports to external file formats, although the can also be displayed “as is” using a pre-output format within the UI. When writing to external devices, the MRG can generate PDF, CSV, RTF, DOCX or HTML files based on a specified stylesheet.

Externally written reports can also maintain the auditing information for future use. An example of this behaviour can be seen here, where the PopPKPD Reporting system has generated a Word document containing “comments” containing the audit information:



FUTURE DEVELOPMENT

We have designed the system such that it is easy to extend. Future extensions under consideration include:

- Relaxing the "NONMEM" requirement so model-based analyses performed using other systems can be reported using the same system. Other systems under discussion include MATLAB and R
- Relaxing the "PopPKPD" requirement so other models type (eg. survival or repeated measures) can be reported using the same system
- Allow the user to actually "run" the NONMEM models via the interface, providing a single portal for NONMEM modeling, monitoring and reporting

CONCLUSION

The system discussed here went live in August 20099, and was supported by user training. The second phase of development is due to start in January 2010

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Richard Pugh
Mango Business Solutions
Unit 2
Greenways Business Park
Chippenham
SN15 1BN
Work Phone: +44 1249 767 700
Email: rpugh@mango-solutions.com
Web: www.mango-solutions.com