

AN APPROACH TO STANDARD PROGRAMMING IN A CLINICAL DATA REPOSITORY

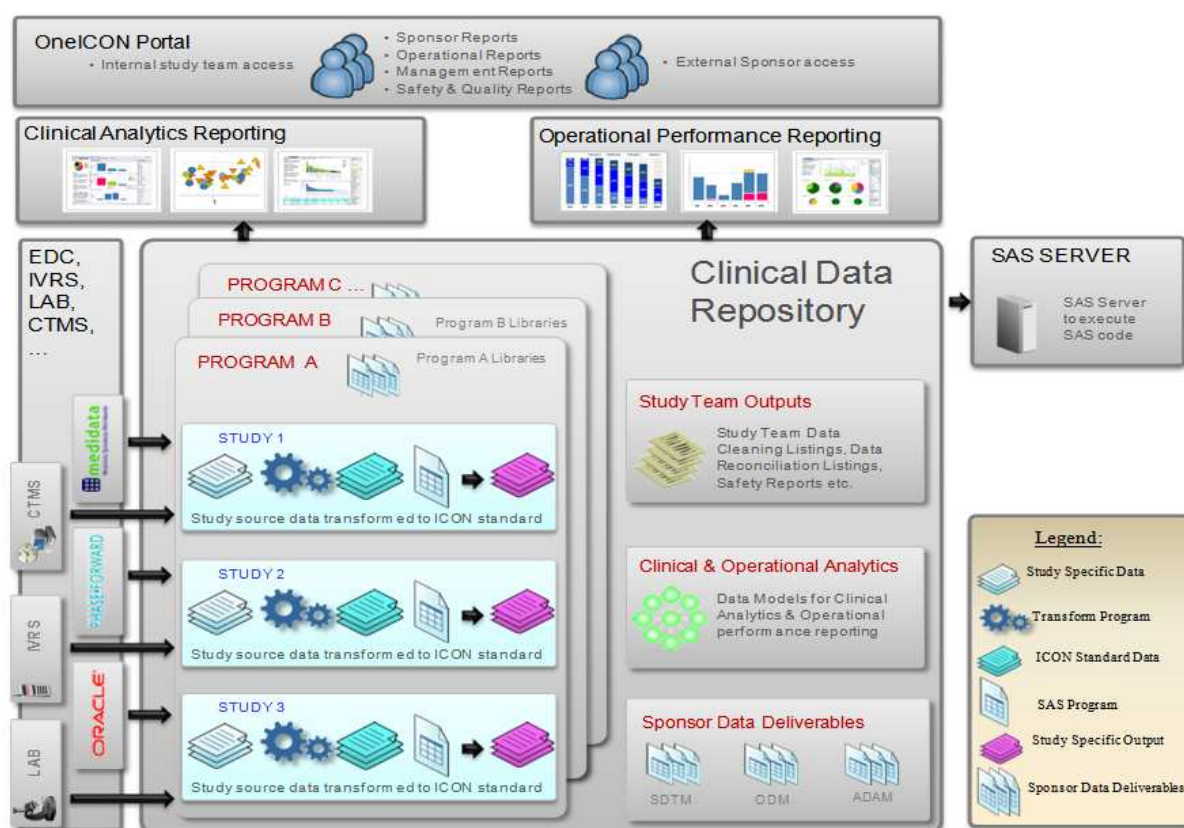
Robert Ellison, ICON Clinical Research, Dublin, Ireland

ABSTRACT (AN APPROACH TO STANDARD PROGRAMMING IN A CLINICAL DATA REPOSITORY)

The purpose of this paper is to share ICON's experiences of setting up and populating a global library of standard programs.

INTRODUCTION

The primary reason for ICON deciding to invest time in implementing a clinical data warehouse infrastructure was to enhance the analysis and delivery of clinical trial data to sponsors. The decision was made in the beginning of 2009 to use Oracle's Life Science Data Hub to deliver a Clinical Data Repository (CDR) as a foundation to support clinical data integration and reporting within ICON. The high level goal was to have one central location in which data from all sources (internal and external to ICON) could be stored, using a global approach to processing and analysing the data. The CDR went live in July 2010



Clinical Data Repository

PhUSE 2011

With the repository and standard data structures in place one area of focus in 2011 has been to develop a library of reusable code. After carrying out an analysis on legacy programs that were being used to produce a range of outputs required to facilitate data cleaning and study metric reporting, it was decided to approach standard code development from two different perspectives:

Generic parameter driven code developed for scenarios where a set of standard algorithms can be developed to produce the desired output. For example, a global standard Adverse Event Report generated from the standard data structures will apply the same logic to produce the outputs for every study. Where study related visit or treatment information is required specific information can be fed into the logic of the programming code using parameters without the need to alter the structure or logic of the program code. Once this code is validated and placed in the global library it can exist as an object in its own right or be embedded in a standard program in the global library or code in a study specific location within the Clinical Data Repository. Validation is not required at an individual usage level.

Generic code templates developed for scenarios where a set of standard algorithms cannot be developed to produce the desired output. For example, the more complex aspects of the transformation from raw data to the standard structure have to be manually coded due to the study specific nature of the data structures. It can be difficult to develop a set of rules that can be applied to non-standardised data so in such scenarios program templates are developed containing generic elements of code and comments where study specific input can be applied. This gives the programmer a standard base to work from and the flexibility to apply specific algorithms where required. The code template is placed in the global library as an unvalidated program or set of programs. To utilize this template it must be copied from the global library area and placed in study specific location within the Clinical Data Repository before the template can be updated. Completed code must be validated at the study specific level before it is moved into a production environment.

SHORT AND LONG TERM BENEFITS OF HAVING A SUITE OF GENERIC PROGRAM CODE IN ONE CENTRAL LOCATION

The short term benefits of having a suite of reusable generic code fall into two main categories.

Reduced effort - Standard code does not need to be developed, reviewed or validated each time it is applied to a study. The effort to program the code and push it to production is a once off exercise for as long as the program requirements stay the same. There are also significant reductions in the effort required to roll out code using pre-prepared templates. Templates will utilise validated standard macros (such as standard date conversions) embedded in the set up sections, partial code with references to standard tables and variables and pre-formatted outputs. This will reduce the amount of development, review and validation effort required.

Standardised approach to programming & formatting outputs – The reuse of standard code ensures consistency in the way the reported information is collated and the format in which it is presented. When using a large global team to program listings and reports for different sponsors and therapeutic areas it can be difficult to maintain consistency in these areas. A carefully controlled and maintained library will allow a large team to share code and develop a standard method of reporting.

In the longer term as the library is further developed and an increasing volume of standardised data and reports are accumulated the ability to report consistently across studies and therapeutic areas for purposes of clinical, medical and statistical analysis will prove invaluable.

LIBRARY MAINTENANCE

Within the CDR a standard library environment has been created to develop and publish standard program code. The development domain was created for programmers to develop objects in a controlled development environment and push these objects through the validation life cycle. Code is created or updated in a development work area where it can point to source development data and place output into target tables or formatted files. Once development is complete the code is then moved to a QC environment within the development domain, where a review of the code and associated outputs can be carried out by another programmer and member of the operational team. In this environment the code can be pointed at production data to carry out the final quality steps. Once the code is ready for use in production it is promoted into the published domain which is a shared environment. Within this domain program code can be referenced from calling programs at a global library or study specific level. If objects are copied from the shared published domain into a study specific location all connections with the original source library are removed and the program can be updated in the new location without impacting on the source objects. This process is used when utilising code templates.

PhUSE 2011

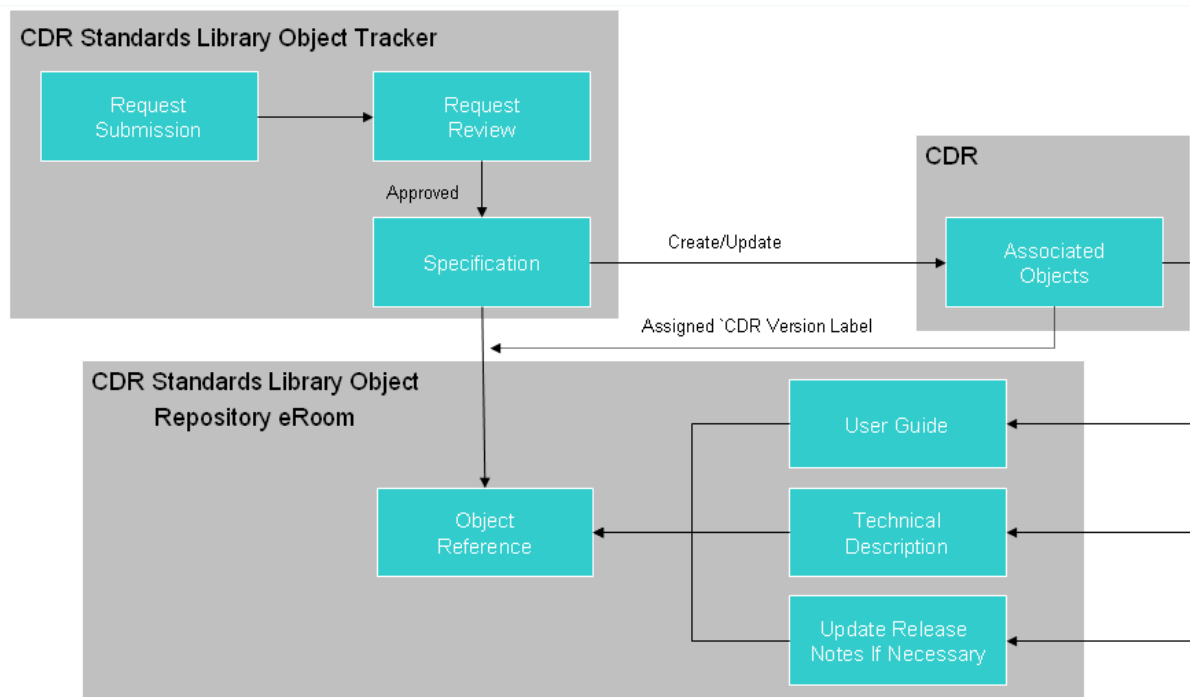
The global library environment is fully version controlled, as are all domains within the CDR, to ensure that changes are tracked on published program code and associated objects. When a change is made to a published piece of code a decision can be made at each of the locations where the code is referenced to accept the new version or opt to remain with the previous version of the code. It is important to have this level of control to ensure that any performance enhancements or process improvements can be recognised at specific locations or can be ignored if there is a possibility this could impact on the required functionality.

The library is only useful to programmers if they can easily find the objects they are looking for and if the objects themselves are maintained. In order to make sure that the code in the library has gone through the necessary development cycle and that the function of the finished product is apparent to whoever is searching the library, two databases with associated processes were put in place.

To track the ongoing activity in the standard library a Standard Library Object Tracker database was created to record the status of each activity under the following categories:

- Request for programming under review
- Request for programming rejected
- Approved pending development
- Published

The programmer developing code in the library will receive a request to develop code for the library or update existing code for inclusion into the library and will enter this information into the tracker to ensure a record of the evolution of objects in the library is maintained. Once a program is published the details relating to the type of activity (e.g. listing program, utility program, etc.), published program versions, specifications versions and current program status are recorded in the Library Object Repository. This Repository shares a unique object reference with data in the Object Tracker and is the tool programmers use to identify code they require, by function, in the library. This process is summarised in the following diagram.



Standards library maintenance

PhUSE 2011

PROGRAMMING METHODOLOGY

For the purposes of demonstrating the approach the ICON DM Programming team apply when creating programs for the standard code library, we will follow a suite of standard lab data reconciliation listings through the development process.

When a programming request is submitted the first step is to carry out an analysis of the programming requirements as outlined in the specification document. This process is followed to see if any patterns in the logic of the programming requirements can be identified and incorporated into algorithms. This reduces the amount of task specific code required and is also used to determine if code meeting the specified criteria already exists in the standards library. This analysis is carried out on all new or updated program code to ensure any patterns identified can be placed into reusable macros or data steps.

Standard Check Name	Study Specific Check Name	Check Description
LAB_01_STD		List any Subject Number [SUBJID] present in clinical database but does not exist in External data file
LAB_02_STD		List any Subject Number present in external data file but does not exist in Clinical database [SUBJID]
LAB_03_STD		List any Site Number [SITENO] present in clinical database but does not exist in External data file
LAB_04_STD		List any Site Number present in external data file but does not exist in Clinical database [SITENO]
LAB_10_STD		Date of lab sample at Visit X in the clinical database does not match date of lab sample at Visit x in the external file
LAB_11_STD		Gender recorded in clinical database does not equal gender listed in external data file for patient x
LAB_12_STD		For US patients only: Date of Birth [BRTHDAT] recorded in clinical database does not equal Date of Birth in external data file for patient x
LAB_13_STD		For EU patients only: Age [AGE] recorded in clinical database does not equal Age in external data file for patient x

LAB_01_STD
LAB_02_STD
LAB_03_STD
LAB_04_STD

Pattern 1: Value from variable A, table X is not in variable B, table Y

LAB_10_STD
LAB_11_STD
LAB_12_STD
LAB_13_STD

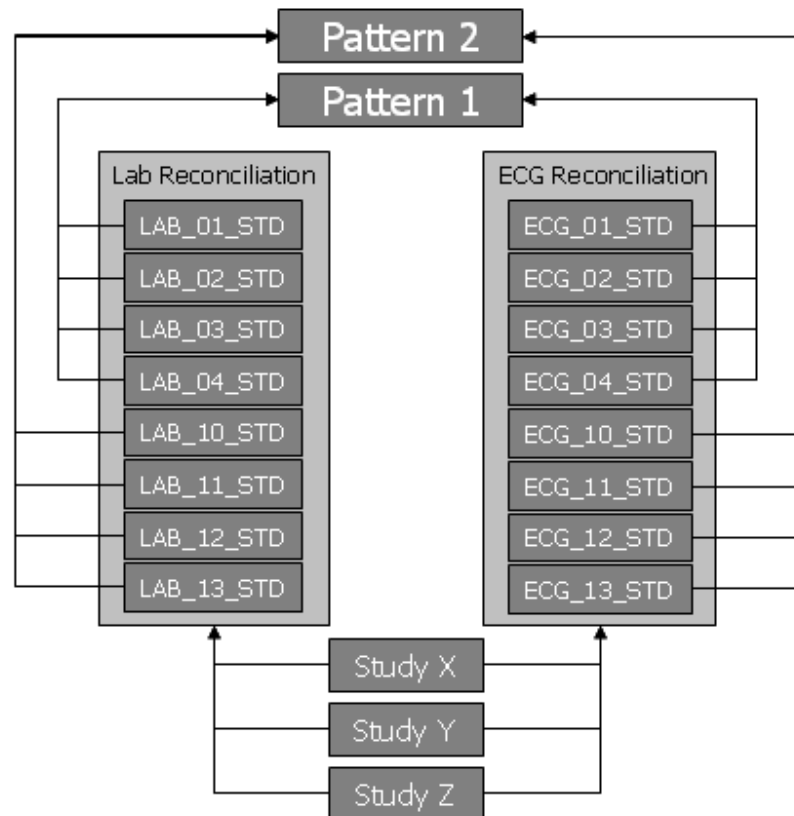
Pattern 2: Value from variable A, table X is not equal to value from variable B, table Y

Pattern Identification in code specifications

The patterns identified within the logic of the program specifications for this example can be translated into two parameter driven macros. The programmer will check the Standards Object Repository to see if code containing this logic exists. If this code already exists in the library then it will be incorporated into a study specific calling program to produce the required output. If the code does not exist in the library then these macros will be developed in the library environment and pushed through the validation life cycle.

Once the standard macros are in place in the published domain they can be utilised for the any study requiring lab reconciliation outputs. Due to the generic nature of the macros that have been developed it is also a probability that they can be used for other purposes, e.g. reconciliation listings for other data types such as ECG information. The macros exist as objects in the library but can also be incorporated into standard program code and/or template code as embedded objects for more specific use.

PhUSE 2011



Applying Identified Patterns

CRO SPECIFIC CHALLENGES - WORKING WITH MULTIPLE DATA STRUCTURES

Working for multiple sponsors and receiving requests for sponsor and study specific outputs means that there are currently limitations in what we can standardise from a programming perspective. Although there are many similar areas of study reporting across different sponsors and therapeutic areas we often have to meet very specific programming requirements which require an element of study specific programming. However, in these instances there are still significant benefits in working from a standard library as the reuse of even portions of code reduces development, review and validation time and promotes consistency.

Our main focus to date have been on standardising and reporting on clinical data. Progress has been made in relation to reporting on study status metrics but the challenge that we are facing is how to represent discrepancy, audit and metadata information coming from multiple Clinical Data Management Systems (and other sources) in a standard and consistent fashion, to facilitate cross study reporting. This continues to be an area of focus as ICON continues to build and improve on our reporting capabilities.

NEXT STEPS - FUTURE PLANS

As we continue to build our standard programming library and develop our ability to produce standard study metrics it will be important for ICON to increase the level of access and control end users have on the outputs we are generating. Reports that are currently generated in this environment are in some instances using Business Intelligence tools to allow users to drill into the details of reports and control how the data is filtered and ordered. Our focus will be on increasing the end user's ability to control what information they see and how it is presented to them. This will allow us increase the number of programmable reports that can be standardised by giving the end user the ability to format the data to meet their requirements.

PhUSE 2011

CONTACT INFORMATION

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

Robert Ellison
ICON Clinical Research
South County Business Park
Dublin 18
Work Phone: 00353 (01) 2912405
Email: robert.ellison@iconplc.com
Web: www.iconplc.com



Clinical Research