

Building Block macro principle for Standard Program Library Design

Francis Dsa, Novo Nordisk A/S, Copenhagen, Denmark

ABSTRACT

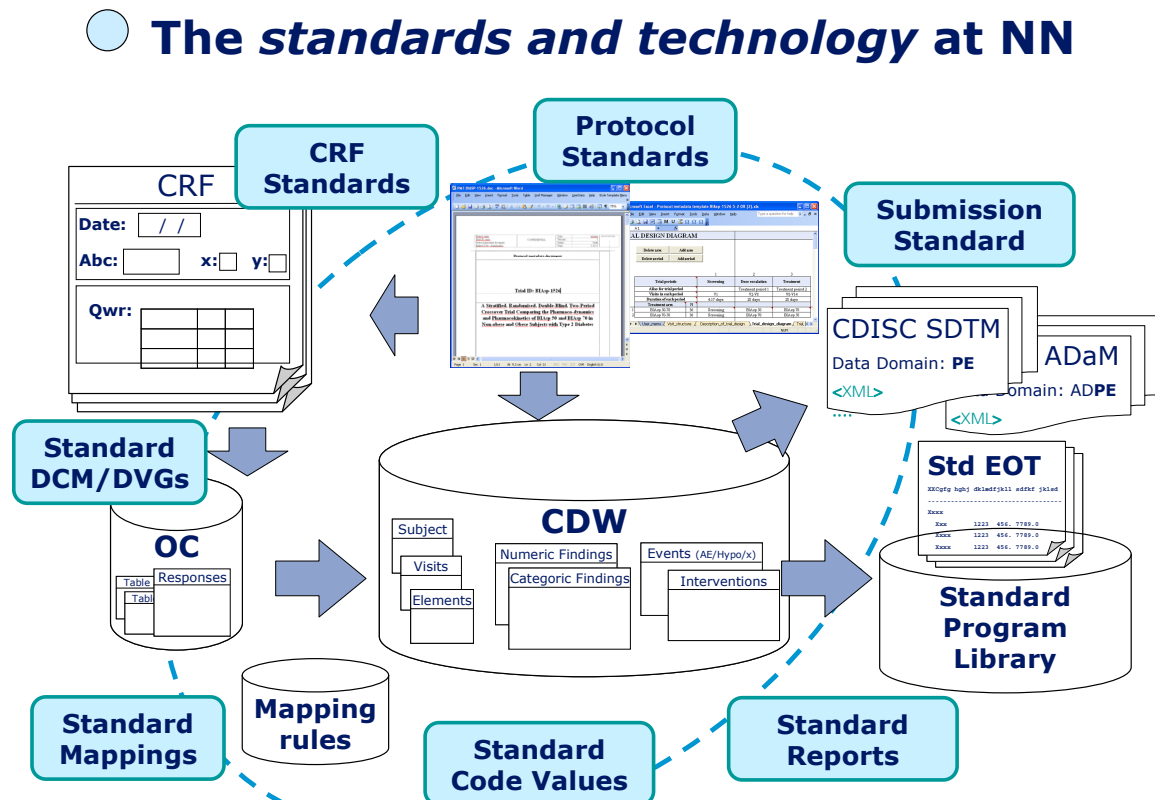
At Novo Nordisk A/S, Standard Program library has been created using the “Building Block macro principle” supporting the analysis and reporting in the Clinical Data Warehouse system. The Clinical Data Warehouse is a 21 CFR part 11 compliant global systems and resides on a common data model based on CDISC SDTM.

In this paper, detail implementation methodology and module based approach used for the development of our library of several SAS standard macros, programs or utilities using the “Building Block macro principle”.

I will also share about the experiences our colleagues in Novo Nordisk and the CRO who were part of the development of Standard program Library about the use of “Building Block macro principle” and discuss benefits and challenges.

INTRODUCTION

The Clinical Data Warehouse (CDW) is a global statistical computing environment supporting standardised data structures and standard programs for analysis and reporting. In addition, the system offers applications to support the mapping and transfer of clinical data and metadata this environment. CDW enables users globally in Research and Development to analyse and report clinical results. On top of the CDW, a Standard Program Library is implemented with SAS components for the statisticians and statistical programmers to handle and analyse the clinical data, report and submit the results.



BUILDING BLOCK MACRO PRINCIPLE:

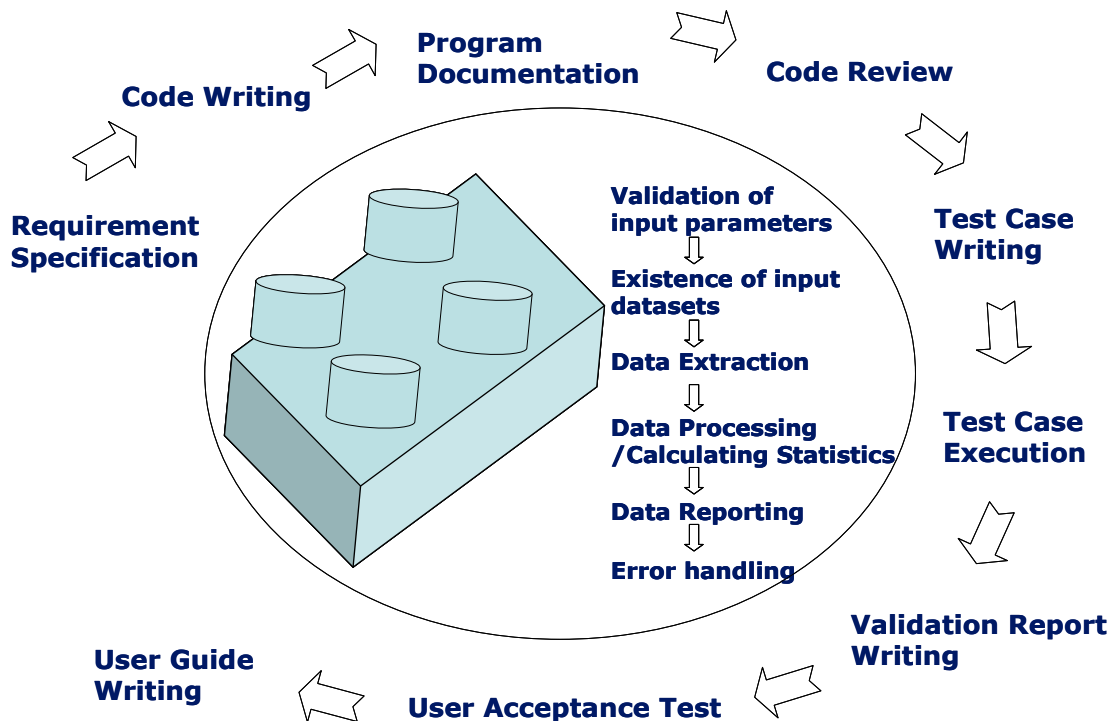
Building block macro (BBM) is basically like a LEGO® brick independently developed following the standard implementation methodology and a module based approach making sure it fits in the bigger picture of developing a Standard Program library ensuring consistency, portability, reusability and easy maintenance.

For analysis and reporting in the Clinical Data Warehouse system, SAS standard components are developed and validated to be used in Novo Nordisk trials across R&D centres, indications and therapeutic areas.

The idea behind this framework is to design sub-components that can be reused in trial-specific programs and across SAS standard programs in order to minimize overall effort in programming, documenting, validating and maintaining the standard program library.

Each standard program component follows the same software development cycle: requirement specification, code writing, program documentation, code review, user guide writing, test case writing, test case execution and validation report writing.

Building block macro principle or the “Lego®” brick principle.



PhUSE 2011

CATEGORIES OF BUILDING BLOCK MACROS:

Building Block macro was categorized as below:

INITIALIZATION BBM:

%access_v01.sas: This macro sets up the autocall libraries for the Graphs, Listings, Macros, Tables and Utilities folders and also sets up the libnames for Data folder and Output folder as chosen by the user. It also sets the global macro variables and the format catalog.

DATA EXTRACTION BBM:

%extract_find_v01.sas: This macro extracts finding data and is filtered according to the selected input parameters.

%extract_event_v01.sas: This macro extracts event data and is filtered according to the selected input parameters.

DATA PROCESSING BBM:

%calculate_catgo_stats_v01.sas: This macro calculates frequency statistics grouped by the by group variables specified in the input parameters.

%calculate_num_stats_v01.sas: This macro calculates statistics and percentiles specified in the parameters grouped by the by group variables also specified in the input parameters.

%calculate_harmonic_mean_v01.sas: This macro calculates the harmonic mean based on a specific numeric variable for each group in a dataset.

%calculate_geometric_mean_v01.sas: This macro calculates the geometric mean based on a specific numeric variable for each group in a dataset.

DATA REPORTING BBM:

%print_summary_v01.sas: This macro formats the input dataset into a form needed to summarize using proc report. The indentation, alignment, page grouping, splitting of lines etc. happens in this macro.

%print_listing_v01.sas: This macro formats the input data set into a form needed for listing using proc report. The indentation, alignment, page grouping, splitting of lines etc. happens in this macro.

%print_mean_plot_v01.sas: The program produces a plot using proc gplot.

DERIVED VARIABLES AND ENRICHMENTS (DVE) BBM:

%mk_age_groups_v01.sas: This macro is grouping subjects according to their ages based on the Business Rule selected in the Metadata.

%mk_dve_partial_date_v02: This macro handles the partial dates based on the Business Rule selected in the Metadata.

%mk_dve_trt_emergent_v01: This macro sets the treatment emergent flag.

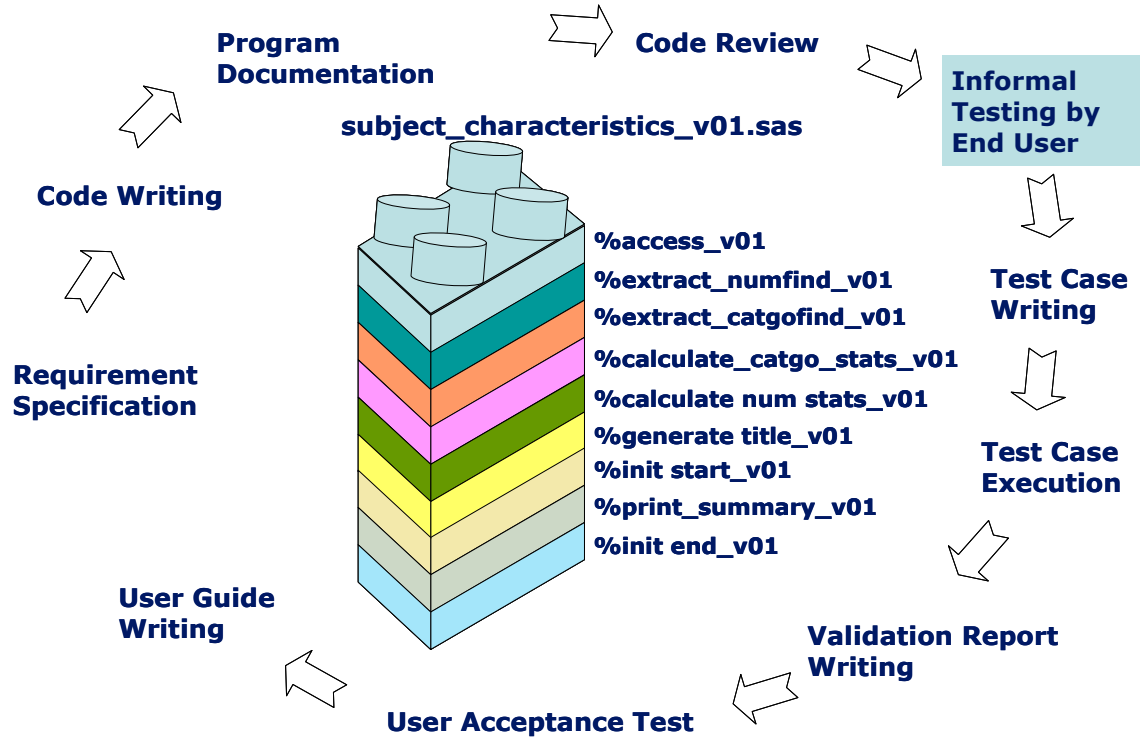
%mk_dve_hypo_count_v01: This macro counts and classifies the Hypoglycaemic Episode.

APPLICATIONS OF BUILDING BLOCK MACRO (BBM) PRINCIPLE:

APPROACH 1:

In the below figure, I am building a SAR program `subject_characteristics_v01.sas` using the Building Block macro principle. I have used several BBM's using the LEGO principle following the same software development cycle. Informal Testing by End User is done earlier in the process ensuring the output meet the Requirement Specification.

Developing SAR (Statistical Analysis and Reporting) programs



BENEFITS:

- Consistent Output across project.
- Reusability.
- No Program Documentation needed.
- Time is saved from routine programming.
- Opportunity to develop standard utilities.
- Time is saved from routine programming.

CHALLENGES:

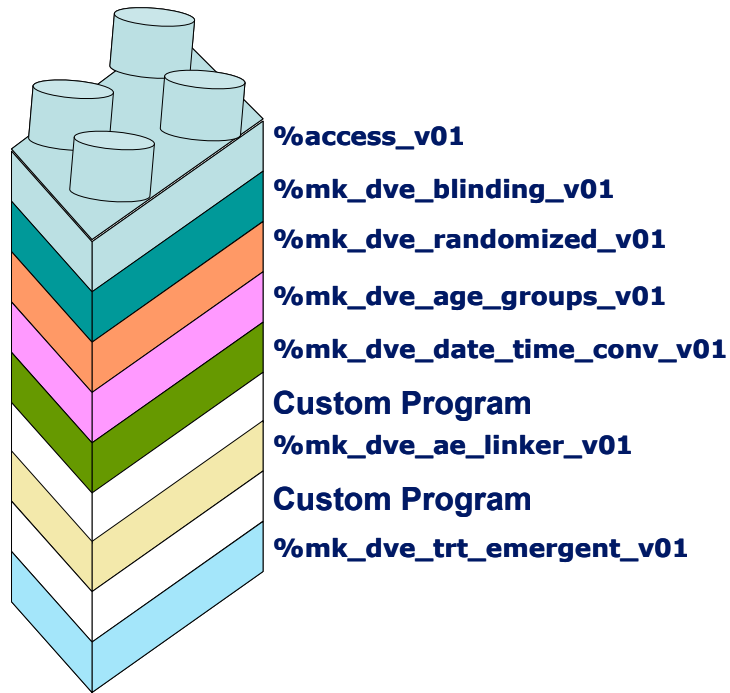
- Lack of Control.
- Custom changes not possible.
- Debugging is a challenge when an error occurs.
- Dependence on the Standard Program Developer team for changes.
- Quick changes are not feasible.

APPROACH 2:

In the below figure, I am building a Derived Variables and Enrichment (DVE) program using the Building Block macro principle. I have used several BBM's using the same LEGO principle. However the outer program is not validated as per the software development cycle with opportunity for Custom programming.

Developing DVE (Derived Variables and Enrichments) programs

Custom Framework for DVE



BENEFITS:

- More control to the trial programmer.
- Users can be creative.
- Less Dependency on Standard Program Developer team.
- Quick fixes are possible in the White Box area.

CHALLENGES:

- Complexity is high.
- Time Consuming.
- Experience is essential.
- More Documentation and Review work.

END USER SURVEY RESULT

It was seen in the survey that

- Experienced users are happy with DVE , less with SAR.
- Main advantage in using standard components is seen as reusability and consistency.
- Main disadvantage in using standard components is seen as lack of control and time consuming.
- Only few SAR building blocks macro are used in custom SAR programming.
- Lack of Trust was seen while using the SAR components.

WHAT WE DO TO IMPROVE THE WAY THAT WE WORK WITH STANDARDS

- Training, Training, Training and also Training!
- Improve toolbox of reference help and user guides.
- Increase Knowledge Sharing through active user forum.
- Feedback and survey from the end user.
- Listening to our end user.
- Job swap of employee.

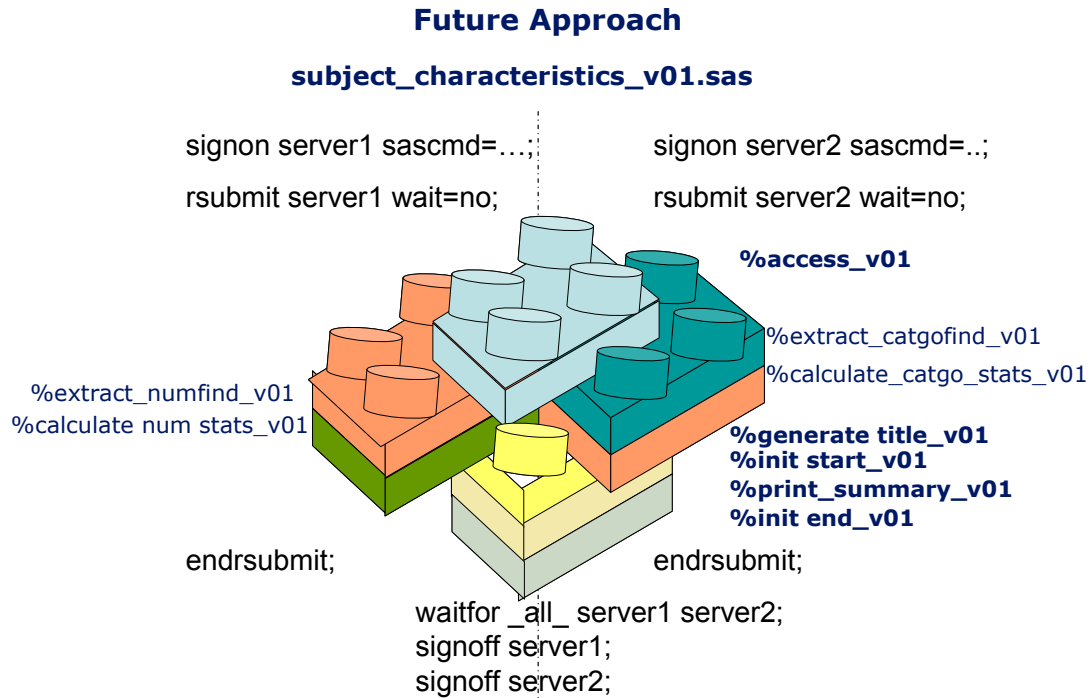
CHALLENGES AS A STANDARD PROGRAM DEVELOPER

- ✓ Development of standard programs is a difficult and time-consuming exercise.
- ✓ Versioning and maintaining of Standard Program and Building Block macro.
- ✓ There is more paper work involved in standards management, e.g. bugs reported in IT system management (Remedy) and change request raised for statistical standards.
- ✓ It can also be difficult to have the necessary resources as clinical projects often have the priority.

THE FUTURE APPROACH:

In the below figure, we are how to build the future SAR Programs subject_characteristics_v01.sas using the Building Block macro

Here I am using several BBM's parallel to make use of SAS parallel processing to best utilize the UNIX processor advantages.



To use the Power of SAS parallel processing in order to best utilize the UNIX box with 32 processor.

PhUSE 2011

RECOMMENDED READING

CD12 Bridging the Communication Gap Implementing Metadata Standards in Trial Protocols
Annabel Acs, Novo Nordisk & Pia Hjulskov Kristensen, Novo Nordisk (Phuse, 2010, Berlin)

AD07 Metadata and Standard Programs
Marianne Carames, Novo Nordisk & Martin Lindhard, Novo Nordisk (Phuse, 2009, Basel)

TS06: Managing your metadata efficiently
Kirsten Langendorf, Novo Nordisk & Mikkel Traun, Novo Nordisk (Phuse, 2009, Basel)

AD03: Data Standardisation, Clinical Data Warehouse and SAS® Standard Programs
Jean-Marc Ferran, Novo Nordisk A/S & Mikkel Traun, Functional Architect,
Pia Hjulskov Kristensen, Business Implementation Specialist (Phuse, 2008, Manchester)

CONTACT INFORMATION

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

Francis Dsa
Novo Nordisk A/S
Vandtårnsvej 114
Søborg 2860
Denmark
Work Phone: +45 3079 0799
Email: frd@novonordisk.com

Brand and product names are trademarks of their respective companies.