



MODEL OF INFORMATION RETRIEVAL FOR DIFFERENT CLASSES OF DOCUMENTS

12th October 2018, Warsaw
Denis Syrov

PAREXEL[®]
YOUR JOURNEY. OUR MISSION.[®]

AGENDA

- Introduction
- Search tools use cases
- Search tools high level overview
- Search tools – details of implementation
- Examples of search tools
- Conclusion

INTRODUCTION - I

Work at least partially is repeatable if:

More than 1 project within the same set of standards:

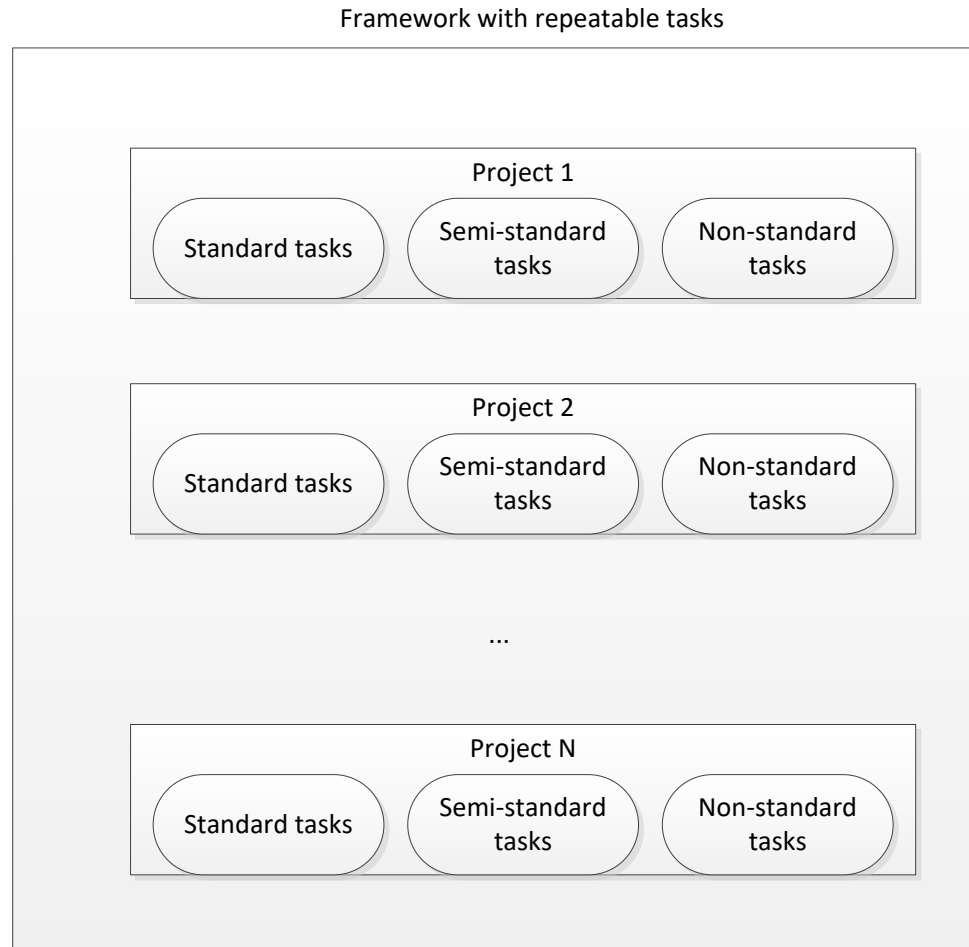
- More than 1 study for the same treatment
- More than 1 study by the same sponsor
- More than 1 study within the same CRO using CRO standards
- Ways to reduce work and gain consistency if repeatable work exists
 - Standard routines for analyses (macro or function libraries)
 - Standard tools for routine tasks (check, collect, create, ...)
 - Central libraries of standard documentation and definitions
 - Search tools for already implemented tasks to inherit from already implemented

INTRODUCTION – II

- If all requirements are available (standard task) – standard tools can be developed. Examples:
 - Table shells have standard set of TLFs where all variability is explicitly shown in shells (layout, parameters)
 - Standard set of CRF pages where all variability is known in advance
 - Standard SAP definitions for parameters
 - Standard names and structure of documentation and folders
- Some tasks may be partially parametrized but not all parameters are known in advance (semi-standard). Examples:
 - Standards are out of control and can be updated by 3rd party
 - Documentation can't be considered as stable enough to be standard
- And there can be completely non-standard tasks. Examples:
 - Rare tasks that doesn't make sense to make standard

INTRODUCTION – III

Semi-standard and non-standard tasks can be tricky to deal with but they can be essential amount of required working tasks:



INTRODUCTION – IV

One of the solutions that can help with semi-standard and non-standard tasks can be development of search tools:

- Search tools can help to find and inherit already implemented solutions – which will reduce required efforts and add consistency
- Search tools can be developed for different types of documentation required for work and can contain any information of interest
- Search itself is standard routing

SEARCH TOOLS USE CASES

Examples and use cases for the search tools:

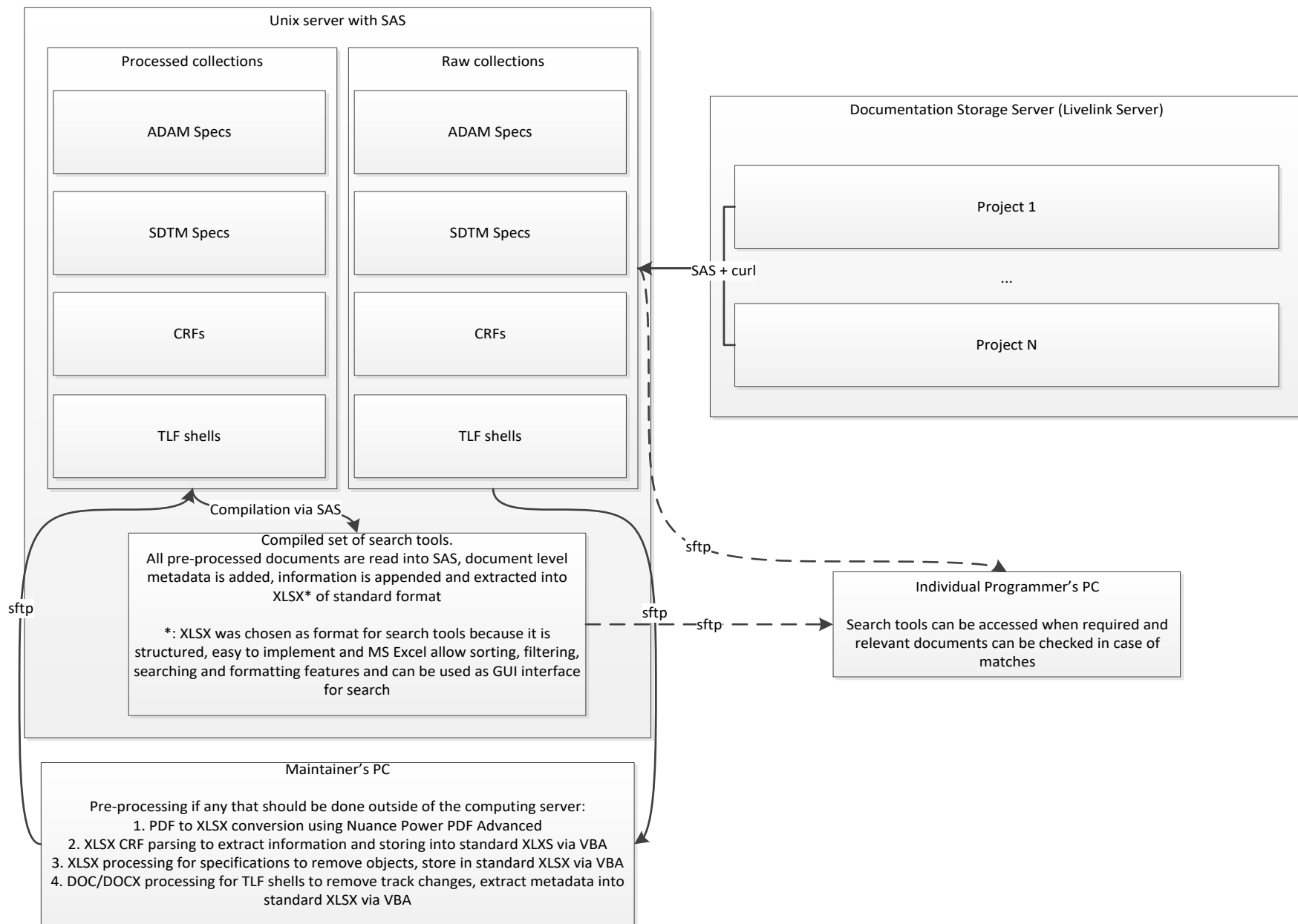
- TLFs can be searched for using metadata (titles, footnotes, programming notes) based on TLF shells and current study can be checked for consistency with previous
- TLFs can be searched and templates can be re-used by Biostatistician for new shells development
- TLFs can be searched for and required code can be inherited
- Relevant CRF pages can be searched for to check SDTM mapping
- ADAM Specifications can be searched for variables/parameters to inherit for specification development
- Programs can be searched for keywords to inherit and use as starting point for new studies

In all cases above required efforts will be reduced and consistency will be added. It will be gained by providing starting point for work that is closer to required result (than inheriting from random study)

SEARCH TOOLS HIGH LEVEL OVERVIEW

- Currently within our programming group we have developed search tools for TLF shells, ADAM specs, SDTM specs, CRFs and SAS programs
- Indexing doesn't require any efforts from study teams
- Indexing is done on regular basis (usually 2 times per month) and all new or updated documents from mentioned classes are collected automatically from all available studies and uploaded into relevant libraries
- Additional automatic pre-processing is done to extract information of interest
- Processed documents are compiled into search tools with addition of document level metadata

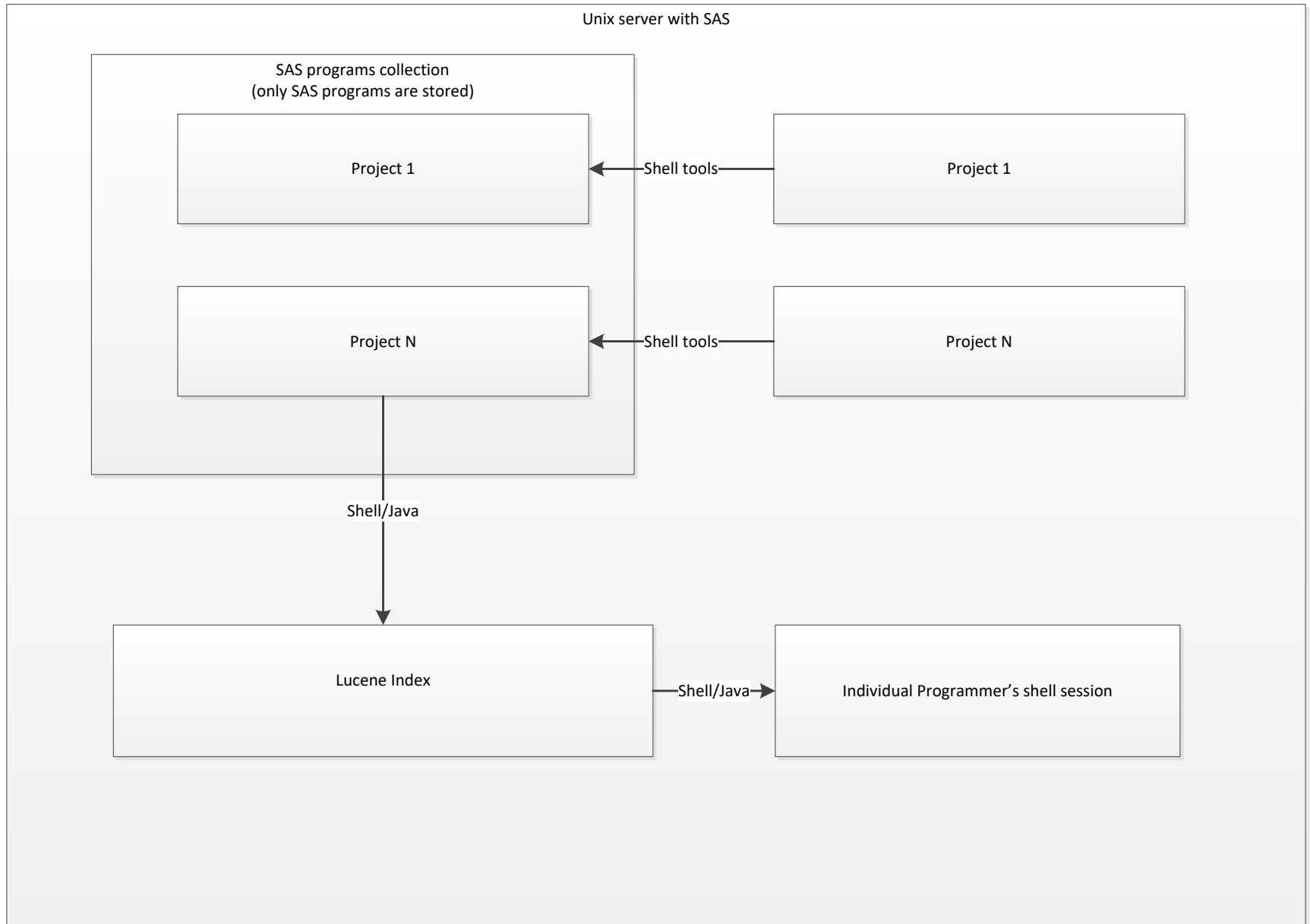
SEARCH TOOLS – DETAILS OF IMPLEMENTATION I



SEARCH TOOLS – DETAILS OF IMPLEMENTATION II

- SAS programs are simple text and no pre-processing is required outside of the server
- Number of SAS programs per project is much bigger than number of documentation
- Search can be done using standard OS tools – in UNIX via grep and find but both doesn't use index and in case of large corpus of programs search query can take >10 minutes
- To speed up search queries for programs we used Apache Lucene which is a high-performance, full-featured text search engine library written in Java - time per search query was reduced to 5-7 seconds

SEARCH TOOLS – DETAILS OF IMPLEMENTATION III



EXAMPLES OF SEARCH TOOLS (SPECIFICATIONS)

- Specifications variable level search tool:

ProjectName	Indication	ServerCode	displayName	modifydate	dsName	paraml	varNm	keyV	varLab	varLe	varFmt	codelist	srcDrv	displayPath
xxxxxx	yyyyyy	zzzzzz	<specification file name>.xlsx	2018-09-06 17:10	<dataset name>	*ALL*	USUBJID	1	Unique Subject Identifier	20			ADDM.U SUBJID.	<path on document server>.xlsx

Allows search across studies based on documentation level metadata (study id, indication, last modification date) as well as variable level metadata (variable name, parameters, code lists, labels, derivations)

- Specifications dataset level search tool:

ProjectName	Indication	ServerCode	displayName	modifydate	dsName	dsLab	dsStr	dsClass	dsDesc	displayPath
xxxxxx	yyyyyy	zzzzzz	<specification file name>.xlsx	2016-03-18 12:42	ADSL	Subject-Level Analysis Dataset	One record per subject	ADSL	Subject disposition, demographic, and baseline characteristics. All Subjects.	<path on document server>.xlsx

Allows search across studies based on documentation level metadata as well as dataset level metadata (dataset name, label, structure, class, description)

EXAMPLES OF SEARCH TOOLS (CRFS)

- CRF page level search tool

ProjectNa	Indicati	ServerCod	displayName	modifydate	pagen	crfVisit	moduleTitle	moduleid	displayPath
xxxxxx	yyyyyy	zzzzzz	<CRF file name>.pdf	2014-12-25 14:58	27	<visit name>	DEMOGRAPHICS	DEMOG001	<path on document server>.pdf

Allows search across studies based on documentation level metadata (study id, indication, last modification date) as well as CRF page level metadata (page number, visit name, module title, module id)

EXAMPLES OF SEARCH TOOLS (TLF SHELLS)

- TLF metadata level search tool

ProjectName	Indication	ServerCode	displayName	modifydate	out_file_name	content	text	displayPath
xxxxxx	yyyyyy	zzzzzz	<TLF shell filename>.docx	2015-09-30 09:53	<output name>	title 1	Summary of Any Adverse Events by Worst CTC Grade	<path on document server>.docx

Allows search across studies based on documentation level metadata (study id, indication, last modification date) as well as TLF metadata level (output name, titles/footnotes/programming notes)

EXAMPLES OF SEARCH TOOLS (PROGRAMS)

- Indexing tool was developed to index 3 fields of the document
- Content of the document
- Path to document
- Date of last modification

As a result this fields can be queried accessing Lucene via shell

Example:

```
progsearch -query 'contents:("adbio") AND  
filepath:("*prog/analysis/*") AND filelmod:("[2016" TO "2030"])' -  
paging 1000
```

Here *progsearch* is alias for Lucene call, **adbio** text (all text is turned to lower case) is searched across all studies in **datasets prog area** and filter for programs updated **>= 2016** year is used

CONCLUSION

- Search is standard routine
- It can be used to efficiently find and re-use of already implemented tasks
- Found results are better than inheriting from randomly selected study
- Search tools can be used as input for other frameworks and meta analyses

THANK YOU

*Denis Syrov
Manager, Statistical Programming
PAREXEL International
Denis.Syrov@PAREXEL.com*