PhUSE 2014

Paper PP37

Technology Trends with the Potential to Change Clinical Development

Chris Olinger, d-Wise, Morrisville, USA

ABSTRACT

The current clinical development process is fraught with anachronisms and outdated technologies. This poster explores some of the trends in technology from an edge-dweller's perspective: leveraging unstructured data models, open source analysis and reporting tools, semantics and metadata driven content, source repositories, and PaaS and Cloud-based deliveries. It should be possible to modernize clinical process development by utilizing these platforms, yet the industry is slow to adopt (or adapt) to them. The outside world is moving forward, and clinical process designers must find a way to move forward as well or forever be reliant upon costly and outdated techniques. The companies that find a way to incorporate these next generation ideas stand the best chance to move their organizations forward.

INTRODUCTION

This poster presentation deals with and defines trends that are rapidly changing the way that companies leverage and deploy technology. We live in a world of cloud based services, remote storage, unstructured data, and Big Data solutions. Yet, the clinical industry is primarily operating using the same techniques, and leveraging the same software stacks as they have been doing over the last 20 years. The biggest events that have altered the way that companies execute trials has been the advent of EDC and CDISC related standards (SDTM and ADaM). However, even these changes are about standardizing what is essentially a paper inspired model. SAS® is still the primary analysis and reporting tool. Servers are still maintained and hosted locally by company IT departments. Storage solutions are typically NAS or SAN deployments. And data is still represented as a square matrix, managed by data management stewards, and analyzed by staff trained in the latest PROC's (even the template that was used to create this paper had sample SAS® code snippets in it).

Given this, there are technologies out there today that have the potential to modify the way that clinical analysts do business. We talk about a few of them in this presentation. In no particular order:

- Distributed Source Repositories
- Open Source Analysis and Reporting
- Unstructured Data Models and Semantics
- Cloud delivery systems and PaaS

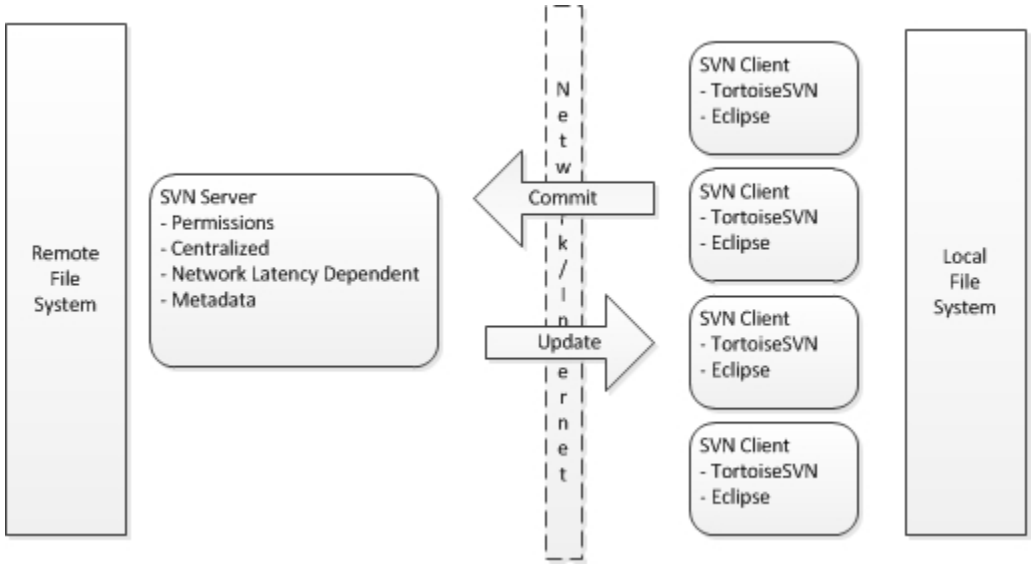
Each of these technologies deserves to be investigated and practical applications created and deployed in the clinical space, even though today they are rarely used.

DISTRIBUTED SOURCE REPOSITORIES

Most clinical companies today leverage some type of source code repository. The grandfather of all repositories is CVS, based on RCS, and is still in use at numerous locations. However, CVS's direct descendant Subversion is more widely used in the Pharma industry. Subversion uses a client/server model where clients pull projects and push changes back to a centralized server. It enforces centralized control of artifacts and allows for direct permissions and metadata on artifacts maintained in the repository.

This model of source control is fairly basic, and typically, Pharma companies use source control in basic ways - to track production versions of programs, outputs, and sometimes data (although data does not function well in source management systems). The client server model is outlined below:

Figure 1



All traffic happens over the network, whether the local Intranet or the external Internet. This leads to one of the downsides of the client server model – performance. In highly distributed setups, such as a repository hosted in a foreign country, it is quite often the case that transfer of assets on both updates and commits is time consuming. SVN allows the user to setup read-only proxy servers to sync with the master repository to deal with this issue, but it would be better if artifacts cold be managed closer to home.

DISTRIBUTED REPOSITORIES

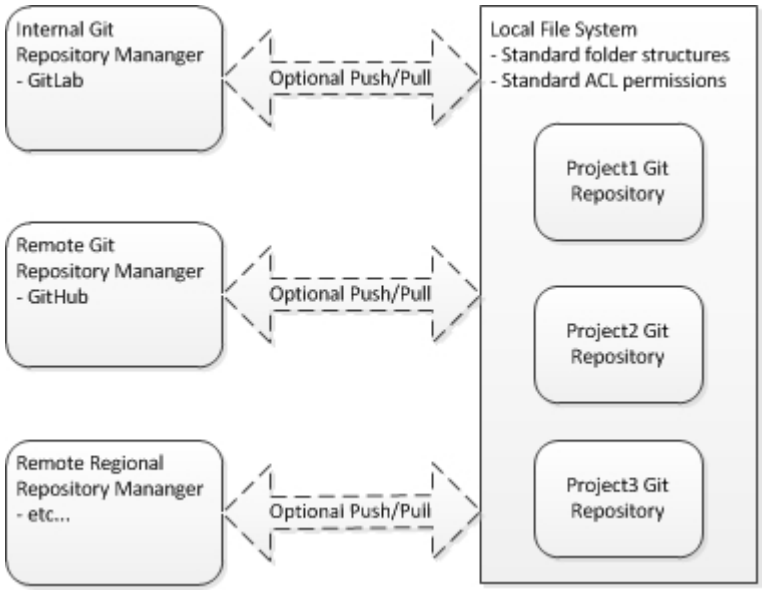
In a distributed source control model, the repositories are not managed under a central server. They are created local to the artifacts that they are managing. As a result, they are more flexible in how you deploy them in remote areas. Commits are tracked locally, and the changes are not merged to a master copy until a user or SOP dictates. The system can be configured to push a repository to a master location, and repositories can be merged to allow flexible creation and destruction of the repositories.

Git and Mercurial are the two most famous distributed SVC systems. In Git, a repository is created and managed in the same folder tree where the programmer is performing work. This repository tracks all changes locally, without ever having to hit a remote server. This paradigm essentially means that the programmer (or programmers) can work locally, recording all changes, and never have to phone home. It also means that an IT department can choose when push and merge local repositories to a centrally managed repository. Users can “branch,” or rather clone, from the local master repository (as opposed to a remote repository). In this scenario, a user has his or her own work folder where a local repository is created that points to the master local copy. All merges back to the master happen onsite.

You may have heard of GitHub. GitHub is a Git repository manager “in the cloud” where people can freely publish (push) their repositories for consumption by the general public. This type of sharing is not feasible for the Pharma industry, so instead of a hosted repository manager there are instead “behind the firewall” options such as GitLab and GerritForge. These services allow for flexible repository management. In fact, the repositories can be pushed and cloned in very flexible use patterns. IT departments can choose how the repositories are managed. No longer is the repository a bottleneck to productivity. Users can track changes and mange work locally, and updates are propagated at defined checkpoints.

Use of distributed source repositories is a boon for controlled flexibility. There use could go a long way towards making source control ubiquitous in an industry that is notably lax in this area. Clinical programmers should use the same tools that software developers have been using for years!

Figure 2



OPEN SOURCE ANALYSIS AND REPORTING

Unless you have been hiding under a rock then you have heard of R. Not only does R have the potential to change how Pharma does statistical programming but it also has the potential to change how results are delivered and how applications are developed. R is an open source language that shares heritage with S, and is very similar to S-Plus. It is completely open source and has a reputation for being a bit of the Wild, Wild West. Current clinical development is heavily SAS® focused, and the majority of BioStats programmers are SAS® programmers. Given this, what are the salient benefits that R brings to the table? And what are the detriments, based on how the Pharma industry does business today?

BENEFITS

As stated, R is an open source statistical programming language. There are benefits to using R for statistical programming. These include:

- R is a lower level language than SAS®. You can easily create your own analysis modules as opposed to waiting on development of new procedures.
- The syntax is consistent, albeit very different from SAS®. R is similar in some respects to scripting languages like Python. As such, it has programming structures that are more modern, such as closures, event driven updating, and object oriented techniques.
- You can leverage modules created and vetted by others. There a LARGE number of modules available for use. The user community for R is very large.
- The development of R is completely open. Bugs are tracked in the open. New features are added in the open. Many users wind up testing the software in the open source community.
- Worldwide resources. More and more R users graduate from University every day.
- Software cost – although this benefit should not be part of the decision equation.
- Graphics support. Graphics are exceptionally well done in R.
- Application development. Shiny is a web application framework that lets you easily and very quickly create web applications that analyze and surface results. It is very simple to use.

These benefits are quite profound. The flexibility of R and the existing code base alone are worth the investigation into how your organization can leverage R.

DETRIMENTS

There is a great deal of uncertainty surrounding R, and its use in the clinical industry. The open source nature of the tool generates questions for QA departments. However, from a Part 11 standpoint, it's about declaring your intent to use, documenting your intent, and then validating and verifying what you have done. In this context, R can certainly be used for analysis, as long as your organization qualifies the tool. Here are some other detriments to the use of R (some real and some perceived):

- Liability and support. The tool is open source. If an issue is found who is responsible? Who fixes it?
- Validation of existing packages. Standard packages need to be validated for use. Where is the clearing house for R packages that are widely in use today?
- Access to source. If you can get it, you can change it. This is very different from the procedure centric model of SAS.
- R is primarily an in-memory tool. Performance on larger datasets has traditionally been problematic.
- R can read XPT files easily. Support for binary sas7bdat files is in its infancy stages. Depending on your data needs you may be unable to process.
- An esoteric syntax – if you are a SAS programmer. SAS lets you do a lot with very little code. R syntax is very different from SAS (but not so different from other scripting languages).
- SAS programmer conversion/training will be required before wholesale departments could switch to R.
- Printed output. SAS reports still handle a wide variety of printing tasks. R's printing capabilities are primarily LaTex based.

As you can see, there are a variety of reasons for and against R. The notable difference is that there is a massive push to add new functionality to the R stack. Over time, some organization will take it upon themselves to become the clearing house for verified and validated modules. Printing support will grow. The Shiny web framework will be enhanced further. It's only a matter of time.

UNSTRUCTURED DATA MODELS AND SEMANTICS

There is a great deal of hype surrounding unstructured data - Big Data specifically - and the analysis of these data stores. In general, clinical programmers never deal with, or intersect with, any of these ideas. Most clinical programmers still deal with SAS datasets, XPT files, and various output types from the SAS reporting procedures. However, unstructured data models present a large opportunity for the Pharma industry. Specifically, unstructured models share the following properties which makes them potentially advantageous for use in the Pharma industry:

- They are schema-less, or at least flexible enough to not require a fixed format for data storage. This does not mean that controls cannot be enforced on structures. It just means that if the structure of your data changes then the data can be updated without costly data warehouse changes.
- They are primarily based on Key / Value pairs, or documents coalescing key/value pairs. This implies that extra information can be added
- The have their own syntax for querying and analytics that is not SQL. It is usually some flavor of query engine that returns similar results as SQL but are usually more flexible.

The ability to add "extra" information is particularly intriguing. Metadata surrounding data values and analysis can be paired with the actual data itself and can flow through systems accordingly. As an example, consider the following columns from a row of data in DM:

STUDYID	USUBJID	AGEU	ARM	ARMCD	BATCH_ID	BRTHDTC
1000	1000-36487-33404	83	YEARS	PLACEBO	G	1000

In an unstructured mode, you are usually representing data in one of three formats:

- Key / Value pairs: for instance, "STUDYID" : "1000" ,coalesced into a "document" with other Key / Value pairs
- A JSON or binary representation of JSON (see below)
- A triple store, ala RDF and the Semantic Web, in either an XML format or Turtle format

In a Key / Value pair model, a document is the set of Key / Value pairs that makes up the document:

```
{       "ID:            :       "1",
        "STUDYID"       :       "1000",
        "USUBJID"       :       "1000-36487-33404",
        "AGEU"          :       "83",
        "ARM"           :       "YEARS",
        "ARMCD"         :       "PLACEBO",
        "BATCH_ID"      :       "G",
        "BRTHDTC"       :       "1000"
}
```

The above representation is JSON format, but you get the picture. The interesting thing about the document is that you can add new Key / Value pairs at any point in time. You can therefore decorate the model with extra information that is carried along with the data. There are a variety of Key / Value stores that are available today. Some interesting ones are DynamoDB, Riak, Cassandra (Column Based), and Mongo. Most of the stores mentioned here are also highly available, linearly scalable, and deploy to the cloud very well. They are designed for applications such as Netflix, Facebook, and Amazon, and are extremely efficient. What we have yet to see are companies utilize these technologies for smaller scale data. This will eventually happen. Data collected in EDC systems can be naturally added to one of these stores, along with any pertinent metadata, in a format that is more aligned with the data being collected. This extra information can travel through an end to end system where SDTM and ADaM are logical outputs.

Semantics provide another way that data can be represented, preserving metadata about the values and analyses. In a Semantic repository, data and information is represented in a Triple Store. A Triple store models basic relationships in the form of "Subject Predicate Object." Each of the relationships in the store reference some attribute definition about an entity. Quite literally, you are binding metadata and information in a structure (Ontology) that describes itself. This information is not lost during analysis, and can decorate the analysis and be used in downstream operations, such as TLF generation. Currently, you are seeing a growing interest in Semantic technologies for the development and communication of data standards in the clinical Industry. These techniques will eventually spill over into a Semantic enabled workflow for clinical - the promise being self-describing data and results.

CLOUD DELIVERY AND PAAS

You have no doubt heard of Amazon's EC2. This service is an elastic cloud computing environment. You can provision resources – servers, network bandwidth, and storage – at a moment's notice, and more importantly, scale these resources as capacity needs increase. This provisioning happens automatically in some instances, and can be controlled very easily through Amazon's EC2 control panel. Amazon (and others) have built out infrastructure that enable software developers and others to abstract out the physical allocation of hardware assets from dedicated internal IT departments to something that can scale based on need. This type of infrastructure abstraction is typically known as IaaS or Infrastructure as a Service. PaaS stands for Platform as a Service, and is one step further towards dynamic application provisioning.

In the clinical industry, a great deal of time is spent installing and validating software stacks. A validated install of SAS can produce reams of paper and add untold costs to your software purchase. Ideally, you would be able to provision a software stack that was already pre-validated. Basically, just ask for a system with software X on it, and it magically appears with the documentation to support it. It should be (and will eventually be) easy to deploy custom cloud implementations of software required for the Pharma industry. PaaS is essentially application infrastructure in the cloud. It is all the pieces that you need to build a SaaS application, already provisioned and waiting to be assembled. In this sense, PaaS is what's known as "Polyglot" and "Polyhost", or rather containing multiple applications, frameworks, and infrastructure that a developer can use to build a SaaS app, and deploy it with ease.

As an example, consider Apache Stratos and RedHat OpenShift. Both are open source PaaS that brings the ability to auto deploy MySQL, Tomcat, and other components for easy application development. They contain support for multi-tenancy, auto-scalable load balancing, Cloud bursting, and other features. Simply put, it is a framework that allows you to build scalable cloud based initiatives. We are not far from the day where custom validated software stacks are quickly and easily developed, deployed, and then retired when they are no longer needed. When this day comes to pass, it will be a boon for the clinical industry, in the form of reduced validation costs.

CONCLUSION

This paper and poster presents trends in the computing industry that have the potential to change how clinical development is performed. These trends are being promulgated through innovative ideas and techniques that have transformed other industries. The clinical industry needs to keep pace.

CONTACT INFORMATION

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

Chris Olinger, d-Wise
1500 Perimeter Park Drive
Morrisville, NC 27560
Work Phone: +1 919 600 625
Email: colinger@d-wise.com
Web: http://www.d-wise.com