

Addressing 21 CFR Part 11 Concerns with Execution and Repository Control

Curt Wehrley, Pinnacle Solutions, Inc., Indianapolis, IN

D.J. Penix, Pinnacle Solutions, Inc., Indianapolis, IN

Wing Lam, Pinnacle Solutions, Inc., Fremont, CA

ABSTRACT

The Code of Federal Regulations (CFR) Title 21 Part 11 represents the FDA's effort to stimulate the use of information technology to achieve improved efficiencies in new drug development, submission, review and approval. While the federal government has opted to provide enterprises with some flexibility as they develop their approaches to compliance, the ruling actually represents another Y2K-type system upgrade effort for the Pharmaceutical industry. Many companies have years of effort and millions of dollars invested in existing software and hardware for clinical trials, and Part 11 touches nearly every system. Achieving compliance, while making the best use of existing systems, represents a major challenge for most enterprises.

This paper will provide a high-level view of a solution to this compliance challenge. The solution described consists of a "framework," built using technologies already utilized in most major pharmaceutical companies, to help enable Part 11 compliance in the area of data analysis. By integrating the functionality of a content management system with the power of SAS [and other] analytic tools, this framework provides repository control and execution control for SAS objects (e.g. data sets, programs, macros, logs, graphs, listings). The solution also offers a single development platform for creating and managing multiple SAS objects, for multiple projects, on multiple operating environments.

INTRODUCTION

Pinnacle Solutions, Inc. is a SAS Alliance Silver partner which has created SAS-based solutions for life sciences companies since its founding in 1996.

Over the last several months, multiple life sciences companies have contacted us for assistance in defining a solution for providing 21 CFR Part 11 compliance for their SAS-based systems. During preparation of our recommendations, we met with several different pharmaceutical and medical device companies to develop a better understanding of how these firms were addressing Part 11 compliance for their SAS-based programs and macros.

In nearly all of the firms visited, we found that SAS is typically one of multiple analytic tools used by the organization for analysis of data. In the typical scenario, a clinical data organization used SAS-based programs for 85-95% of their analytic software needs. The remainder of their analytic needs were addressed by objects based on other analytic tools, including – but not limited to – Excel spreadsheets, S-PLUS, SPSS, and Minitab.

None of the companies visited had a fully-functional solution for Part 11 compliance for SAS-based programs, let alone a solution that could handle other analytic tools. Yet most had in-house access to the one tool that could provide the solution: a content management system, or CMS [typically Documentum].

We sought to create a prototype system which utilized SAS Integration Technologies and Documentum's out-of-the-box functionality to provide Part 11 compliance for SAS-based objects and objects based on other analytic tools. Although the goal of the initial pilot project was to utilize Documentum as the CMS tool of choice, we also sought to develop a design that could utilize other CMS products. By leveraging other existing corporate content management systems, we believe that clinical data

organizations can share and integrate their software tools and data with other divisions of the company.

In the interest of object re-use, we wanted to create a solution that could be adapted by other life sciences companies which have diverse applications and tools working on multiple platforms. The best solution from this perspective would provide flexibility and maximize utilization of existing corporate system investments and infrastructure.

Finally, our development team figured that this prototype development project needed a name. We browsed a Latin dictionary, and found a word that seemed quite fitting to the situation:

"modero" – to keep within bounds; to regulate, restrain; to control, govern, direct

BACKGROUND

BUSINESS ISSUES

Money is tight

Today, nearly all drug company pipelines are relatively meager, in part because the pharmaceutical sector has picked most of the low-hanging fruit in drug discovery. Patent expirations for most of the high-revenue drugs which drove the industry's growth in the 1990s have opened the door for generic drug competition, and eroded both market share and profits for many of the original brands. In addition, company valuations are down as the stock market has experienced a dramatic drop from its early 2000 peak. As a result, drug companies are under increasing pressure to reduce costs, while finding ways to bring viable new compounds to market more quickly.

Statisticians hard to find

Ask almost any manager of statistics at a company that does a lot of number crunching and you'll probably hear a common message: Demand for professional statisticians and analysts has not let up. The stats folks are a fairly scarce resource. Why? Enhancements in technology have led to a rapid increase in the amount of raw data available for analysis, and the amount and types of data in corporate databases is expected to continue growing at an exponential rate of increase.

The demand for statistics expertise will probably continue to increase, and companies will likely be forced to make better use of a statistician's time and their existing analytic software tools.

TECHNOLOGY ISSUES

Much has been discussed in the business press about the rise and fall of technology-related spending from the mid-1990s to today. In short, companies in several industries arguably spent too much on IT in the late 1990s to (1) exploit quickly-evolving Internet and other IT technologies, and (2) fix the Y2K problem. Rising prices of company stock prices during the same period only encouraged further spending on enterprise-wide systems and point solutions of many kinds.

Many pharmaceutical companies have learned, however, that buying software presents its own problems. One pharmaceutical company director reported to Bio-IT World magazine (Uehling,

2002) last year, "The software you buy is often generic, and so has little competitive advantage for you. You spend most of your time customizing it to your organization." Like this director, many companies have begun to adopt a more modular approach in which all of a company's software tools run on a network using standard file formats. These kinds of companies believe that they cannot find one vendor who can supply all the pieces they want with all the variety they want. Instead, they have selected the best pieces from multiple vendors for their situation.

Today, most large life science companies are faced with a complex combination of enterprise systems, commercial point solutions, and custom in-house software. And with smaller IT budgets, their CIOs are often experiencing increasing pressure to deliver on the investment made in new systems in the past few years.

REGULATORY ISSUES

Recent SUGI papers and other sources have provided a thorough overview of the key factors for compliance with 21 CFR Part 11, so this paper will not include a laborious explanation of the guideline. Here is a brief summary* of the top areas for compliance that pertain to SAS-based software programs used for data supporting the production of a pharmaceutical drug:

Data security – Only authorized personnel should have access to data analysis programs and corresponding documentation.

Audit trails – For all records utilized in the application of a system, there must be a secure, time-stamped audit trail generated by the system. The audit trail should list all actions taken to create, modify, and delete analysis objects. When a change is made to an analysis object, the system must save the old version and keep track of the date, time, and ID of the person who made the change.

Version control – There must be revision and change control procedures in place.

* - The reader who seeks more detail on 21 CFR Part 11 is advised to review the list of references and the guideline itself.

The remainder of this section will describe pertinent recent observations the FDA's perspective on enforcement of Part 11, and some potential consequences of noncompliance.

FDA Inspectors not e-everything

Pharmaceutical industry observers have said that the majority of FDA inspectors are not yet comfortable with electronic data or documentation. For the most part, these inspectors are scientists, not information technology professionals. As a result, inspectors are more prone to seek hard evidence of, say, an audit trail in a company's applications (Reimers, 2003). In one case, an Iowa-based life science company – which had gone 100% electronic with its data and documentation – was forced by a federal court order obtained by an FDA inspector to print out all of its documents. The resulting paper filled approximately 600 cases (Salamone, 2002).

FDA Shift from Enforcement to Guidance

The FDA has recently withdrawn its previous Part 11 guidance documents and published new, non-binding enforcement plans. The agency has reported that it currently doesn't plan to take regulatory action to enforce the validation, record retention, audit trail, and record-copying requirements of Part 11. The agency also won't take regulatory action against legacy systems (i.e. computer systems operational before Part 11's effective date). Industry observers have interpreted from this a shift at the agency from enforcement to a narrowing of focus and an increase in guidance. The agency has also shifted Part 11 implementation leadership from the Office of Regulatory Affairs to the Center for Drug Evaluation and Research (CDER).

While the agency may be cutting back on its Part 11 enforcement plan, firms are not getting a free ticket to ignore these regulations. The FDA does continue to enforce some of the rules for open and closed systems and electronic signatures.

CURRENT STATE

In the clinical trials data analysis area, many major pharmaceutical companies have built their analysis strategy around the SAS system. Over time, product and project teams in many of these large companies have worked independently to create custom analysis and reporting tools to fulfill the team's analysis and reporting needs.

This homegrown software often means hundreds – even thousands – of large and small programs and macros in multiple physical locations and operating systems throughout the enterprise. Such an environment has produced the following common scenarios:

- Inefficiencies from "reinventing the wheel" – multiple programs in multiple locations that perform similar functions;
- Manual processes are often implemented for promoting objects from "development" stage to "production" readiness; and
- Custom-built in-house application or the native operating system used to control security and versioning.

Years after a study submission, when the FDA or a company employee attempt to conduct an electronic audit trail on the original analysis results and data, it is nearly impossible to find the exact combination and versions of analysis objects and data sources utilized to create the submission document. While an archive snapshot can easily recreate the system, this is often undesired and it can take a long time to reproduce in a "quarantined" environment that is isolated from the current system. (One author has heard of an instance where an FDA inspector asked a programmer to reproduce a set of SAS outputs that had previously been created as part of a submission filing. Under the observing view of the inspector, the programmer was required to recall from the archive the SAS programs used to create the original output set, and re-run the programs to produce an identical copy of the output.)

In summary, most of the companies we visited are working under the following conditions:

- Very limited budgets;
- Multiple technologies and analytic packages – and a desire to make the most of each;
- Multiple operating systems and hardware platform configurations;
- Limited statistical analysis expertise; and
- Constant threat of penalty for CFR Part 11 noncompliance in a time of shifting FDA priorities and uncertain guidance

AVAILABLE SOLUTIONS

The SAS tool set currently does not provide an automatic version control capability (Williams, 2002). This is because SAS programming is not performed within a formal Integrated Development Environment (IDE).

There are tools already available for providing some level of version control to SAS programs and macros. When seeking recommendations for a version control solution from a SAS online discussion group, one typically derives one of three alternatives:

- Microsoft Visual Source Safe – a commercial package;
- Concurrent Versioning System (CVS) – an open-source solution; or

- Revision Control System (RCS) – an open-source solution

From recent SUGI papers, one may read about a professional at a major contract research organization (Williams, 2003) who has described an impressive web-based programming environment for SAS programs.

Each of these tools has strengths and limitations, which have been defined in previous papers. Unfortunately, none provided the full functionality, ease-of-use, and compliance sought by our clients.

At first glance, we thought that the SAS Drug Development solution may be the answer we have sought. However, at the time we began our development of a pilot-size solution, the Drug Development solution was very new and appeared to offer much more than our clients needed. We also felt that the investment required for this option could significantly compromise the current budgets of all but the largest pharmaceutical companies.

The authors of this paper could not find a complete solution with all of the following attributes:

- Customizable;
- Easy-to-use;
- Leverages existing corporate tools and technologies;
- Facilitates migration of existing legacy systems into the new solution;
- Fully compliant with Part 11 requirements for version control, audit trail generation, software management; and
- Accommodates several types of analytical tools and/or different types of CMS products.

IDENTIFYING SOLUTION COMPONENTS

We realized that we needed something that could better automate the process of applying version control and create a traceable audit trail for multiple types of analytic objects. Yet we knew that creating that kind of functionality from scratch would not only be cost-prohibitive, but would also put our clients in an undesirable position of relying on a newly-built, 100% custom, unproven solution for an issue – regulatory compliance – that can keep the most confident managers awake at night.

In order to help our clients gain maximum management and user support for a solution, we felt that our system needed to do more than support regulatory compliance; the system needed to

- Make maximum use of existing technology tools;
- Be transparent enough that it didn't feel like a burden to users; and
- Improve productivity and/or effectiveness in some way.

We found our answer in a product that almost every major life sciences company uses for managing typewritten documents: a content management system (CMS).

REPOSITORY CONTROL USING A CONTENT MANAGEMENT SYSTEM

The need for document management arose in the early 1990s as companies began storing more of their documents on computer systems. Pharmaceutical companies originally saw a document management system (DMS) as a way to automate the process of creating a New Drug Application (NDA), which typically produces thousands of pages.

As the DMS market matured and new technologies – such as multimedia and the Internet – evolved, major DMS vendors expanded the capabilities of their products to manage content built for these new formats. Surviving enterprise-scale DMS companies have become makers of content management systems (CMS), which typically include DMS, digital asset management (DAM), web content management (WCM), and

learning content management (LCM). Today's integrated enterprise content management solutions will allow users to manage and deliver almost all kinds of digital corporate content using one common content platform and repository.

As our team reviewed the DMS features of a CMS, we saw plenty of out-of-the-box functionality that we wanted to apply to the analytic object world:

Security

The DMS applies security rules to each document. These rules, typically defined by the system administrator, define who may create, read, revise, version, delete, or otherwise manage individual documents. Users are required to provide username & password for system access, and all user activities are typically recorded in an audit log. This creates for each document a trail of accountability, and prevents an accidental or intentional deletion of an important document.

File Format Neutrality

As most users of information technology know, electronic documents come in many different flavors:

- Different native applications (e.g. MS Word, MS Excel)
- Different versions of each application
- Different platforms (e.g. MS Windows, UNIX)

A DMS will store electronic documents from all sources, and with little regard to their internal contents.

Versioning

When you revise a file on your PC and save it, the new version usually overwrites the old version, causing that previous version to be dumped into your system's trash bin (and eventually lost forever). Rather than overwrite an old version of a document, a DMS versions the document and stores the new copy alongside the old one. The user who retrieves the document will see only the latest version, but the DMS allows the user to go back [if necessary] and retrieve older versions. This provides a means for recording the entire lifecycle of a document.

Centralized Storage

Most major CMS systems are server-based. This means that they store files on a centralized server rather than on each individual user's PC. The centralized storage location is frequently referred to as a document repository.

Check-in/Check-out

By enforcing a check-in/check-out system, the DMS allows only one person at a time to edit a document. Other users can view the document while it is checked out by the person doing the editing, but those other users cannot edit the document themselves.

Document Attributes

In addition to a document's content, a DMS maintains different items of information associated with the document. These items – examples include title, author, and document creation date – are referred to as metadata.

Full-text Indexing

A DMS provides the ability to search among stored documents with different file formats.

When comparing this list of standard DMS features to our list of required functionality for regulatory compliance, we found that an enterprise CMS software package could provide some very helpful SAS object lifecycle management, workflow capabilities, and repository control. In other words, integration with the DMS portion of a CMS would allow us to define the required stages in the life of a SAS program, establish the process that the program must go through before it advances from one stage to another, and centralize & control the storage of programs. If designed

appropriately and implemented effectively, we believed that this kind of functionality would greatly enhance the productivity and effectiveness of the SAS developers across a life science organization.

Figure 2 provides an example of SAS program lifecycle and workflow in a life sciences organization. (Note that the lifecycle and workflow shown applies to SAS programs before they reach the production stage.)

SELECTION OF CMS

As one might expect, our choice of a CMS for the prototype solution was largely dictated by our clients' current environment. The total investment in a new enterprise-wide content management solution can approach seven figures, so we always want to look first at making the most of whatever CMS system a company has in place. In our case, we found Documentum being used at the site of our planned pilot implementation and at most other life sciences firms we visited, so our CMS research was driven toward that product.

Ideally, we wanted to build a prototype with the flexibility to accommodate any major CMS system on the market. We thought that the best step in that direction would be to build a prototype based on a CMS system being used by a significant portion of the life sciences community. Selecting a mainstream CMS as the foundation for our prototype solution would provide the following benefits:

- A product with an established track record in a regulatory environment and the life sciences market would ensure that our solution was built upon a foundation that would likely be around for years to come
- Choosing a company with a significant portion of the life sciences market would enable re-use of [at least] portions of the solution we prototype, thus reducing costs and implementation time for companies who employ any or all parts of the solution.

Documentum's CMS fits this criterion quite well. The company's technology has been used for years by large pharmaceutical companies. Last year, the company claimed all of the top 25 pharmaceutical companies as customers, and 42 of the top 50. Documentum usually collaborates with similar solutions from companies like IBM and HP, so its CMS is very accommodating for an enterprise-scale system. The company has also improved compliance capabilities of its latest release (version 5), which will let pharmaceutical companies add digital signatures to their Documentum environment.

For creation of the Modero prototype, we utilized Documentum version 4.2 as the CMS software, and Oracle as the database for Documentum.

EXTERNAL ACCESS TO THE SAS SYSTEM

SAS® Integration Technologies

SAS Integration Technologies utilizes industry standards for object transport, message-oriented middleware, and enterprise directory services to provide a standardized common middleware platform and object interface to the main modules of the SAS system. Using this group of open technologies, application developers can use a variety of programming languages and development environments – including Visual Basic, Java, and C++ – to build client programs that access SAS services and objects through a common object interface.

SAS Integration Technologies consists of three groups of middleware technology standards:

- **Component Platforms** – provide support for programmable object interfaces to SAS services

- **Application Messaging Platforms** – provide interfaces to commercial messaging technology from IBM (MQ Series) and Microsoft (MSMQ)
- **Enterprise Directory Platforms** – provide support for directory services such as LDAP and Active Directory Server

In building the Modero application, we used Component Platforms [via an Integrated Object Model (IOM)]. Essentially, IOM is SAS' distributed object interface to the SAS server. Figure 1 shows an overview of the IOM object hierarchy. This IOM is surfaced via the IOM server, a separate process similar to SAS/CONNECT's ® spawner or TCP daemon which facilitates and manages the startup of client-to-server TCP/IP conversations.

The root object in IOM is the SAS ® Workspace, which is the approximate equivalent to a SAS session. Put simply, execution of various SAS services can be accomplished with a Workspace defined in IOM. As shown in Figure 1, once a root Workspace object has been created, other classes are available under the Workspace, including...

- **DataService** – offers Libname assignments and access via JDBC
- **FileService** – allows Filename assignments
- **Utilities** – allows access to HostSystem functions, SAS System options, access to SAS formats in the Java context, and access to the ResultPackage service (including HTML results)
- **Language Service** – allows programmers to submit SAS language statements directly to the workspace or to run SAS StoredProcesses [specially configured SAS programs stored in a file] to allow parameters to be passed to the program via Java

Application

We recognized that we had access to a number of different tools for building a stand-alone application to represent a prototype system. As previously mentioned, SAS Integration Technologies enables an application developer to use an assortment of programming languages and development environments, including Visual Basic, Java, and C++. We found at least one of these tools being used extensively in the larger life sciences firms we've visited in recent months.

For the Modero pilot, we selected Java for its cross-platform compatibility. The application was built using Eclipse – an open source, Java-based, extensible development platform [selected because of its flexibility and visual appeal].

PUTTING THE PIECES TOGETHER

The Modero solution acts as a stand-alone, Java-based client application which provides three primary components:

- **User Interface (UI)** – collects user information/selections and presents results
- **Analytic Services** – business functions layer built on top of IOM
- **Repository Services** – business functions layer built on top of Documentum API

An overview of the Modero prototype's architecture is shown in Figure 3.

Connecting the Application with SAS

A Java client application that accesses an IOM server uses the Common Object Request Broker Architecture (CORBA). CORBA is architecture for an open software bus on which objects can interoperate across networks and operating systems.

The Object Request Broker (ORB) is the key element of CORBA. It provides the infrastructure for distributed object computing by enabling software components (objects) on one machine to locate and communicate with components on other machines.

SAS Integration Technologies provides an ORB called the IOM bridge for Java. This ORB implements the standard CORBA ORB interface defined by the Object Management Group (OMG). It uses a proprietary inter-ORB protocol (called the IOM Bridge protocol) to communicate with the IOM server.

Modero collects information from the user, builds a query or other set of commands, and sends those commands through the IOM bridge for Java (a SAS proprietary ORB) to the SAS tier for processing. The following code is an example of how to submit code to the SAS system:

```
public void submit( String code ) {
    ....
    IWorkspace sasworkspace =
        SASWorkspaceManager.getInstance().getWorkspace();
    ILanguageService saslanguage =
        sasworkspace.LanguageService();
    /* setup connection point to receive execution status*/
    if ( !m_bInitCP ) {
        org.omg.CORBA.IntHolder handleHolder =
            new org.omg.CORBA.IntHolder();
        ConnectionPoint cp = getConnectionPoint( saslanguage );
        cp.Advise( m_submitListener, handleHolder );
        m_bInitCP = true;
    }
    saslanguage.Async( true );
    saslanguage.Submit( code );
    ....
}
```

Access to SAS Datasets

Java Database Connectivity (JDBC) is a common database interface that allows you to access virtually any tabular data source (such as databases, spreadsheets, or flat files) in a standard way using the Java programming language. Typically, a Java programmer can request a connection with a database by using the JDBC application programming interface (API), then send query statements using SQL and receive the results for processing. JDBC handles the connection, sending data and queries to and from the database.

Modero uses the JDBC API to access a SAS dataset through the IOM server. The following code is an example of how to read a SAS data set from a Java client application:

```
public ResultSet getResultset ( String dataSetName ) {
    IWorkspace sasworkspace =
        SASWorkspaceManager.getInstance().getWorkspace();
    ILanguageService saslanguage =
        sasworkspace.LanguageService();
    IDataService dataSetService = sasworkspace.DataService();

    Properties prop = new Properties();
    MVAConnection sqlConnection =
        new MVAConnection(dataService, prop);
    MVADatabaseMetaData metaData =
        new MVADatabaseMetaData(sqlConnection);

    Statement stmt = sqlConnection.createStatement();
    // get all observations of a dataset
    String selectStmt =
        new String( "select * from " + dataSetName );
    ResultSet rs = stmt.executeQuery( selectStmt );
    return ( rs );
}
```

Connecting the Application with Documentum

Documentum version 4i provides an extensive set of APIs called Documentum Foundation Classes (DFCs) that allow a developer to get deep access to the core services in the product. Documentum provides the DFCs in the form of Java classes that run in any J2EE standard application container. Developers can directly access the Java classes in their Java 2 Enterprise Edition applications.

Besides the underlying foundation classes, 4i includes an adapter technology called the eConnector for JDBC, which permits

interaction with Documentum repositories, eContent Server [which manages content across distributed content repositories], and WebCache [provides a mechanism to schedule the publishing and expiration of documents in the 4i repository for web-specific publishing requirements].

Modero collects information from the user, builds a set of commands, and sends those commands to the DFCs, which interpret the commands and send them to the Documentum tier for processing. The following code is an example of how to save a SAS data set to the Documentum repository:

```
void createSAS_Dataset(IDfSession sess) throws IOException {
    try {
        IDfSysObject sysobj =
            (IDfSysObject)sess.newObject("SAS_dataset");
        sysobj.setObjectName("AdverseEvent_dataset");
        //Set attributes
        sysobj.setSubject("sample document");
        sysobj.setContentType("binary");
        sysobj.setTitle("Advers Event dataset");
        sysobj.setFile("C:\\modero\\AdverseEvent.");

        sysobj.link("Study_A_Cabinet");
        //Assume content file and cabinet exist
        sysobj.save();
        //Save
    } catch (DfException dfe) {
        System.out.println("\n" + dfe.toString());
    }
}
```

CONCLUSION

Through development of a pilot system, we have found that it is possible to create a CFR Part 11-compliant environment for SAS-based objects using Documentum and SAS Integration Technologies. In general, we believe that integrating a CMS product with an analytical system can produce several beneficial capabilities:

- Real-time execution and control of analysis objects;
- Traceable audit of objects (tree-view relationships);
- Integration with ECM solutions;
- Multiple platform support; and
- Meta data support for re-usability of code.

REFERENCES

- Cohen, Barry (2003), *Using AppDev Studio and Integration Technologies for an Easy and Seamless Interface Between Java and Server-Side SAS*. Proceedings of the 28th Annual SAS Users Group International Conference, Seattle, WA.
- Depompa-Reimers, Barbara (2003), *Easing the Pain of Part 11*. Bio-IT World, 4/15/2003.
- Guidance for Industry, 21 CFR Part 11; Electronic Records; Electronic Signatures Scope and Application*. FDA, February 2003. www.fda.gov/cder/guidance/5505dft.pdf
- Jenisch, Steve (2000), *SAS Integration Technologies Overview*. SAS Institute white paper.
- Salamone, Salvatore (2002), *IT Needs to Do Its Part for Compliance*. Bio-IT World, 12/10/2002.
- SAS® Integration Technologies Overview
www.sas.com/rnd/itech/papers/oviewSUGI24.html
- Stevens, Jay L. and Santucci, Brian (2001), *Integrating SAS with an Open World: Java, JSP, LDAP, and Oracle*. Proceedings of the 26th Annual SAS Users Group International Conference, Long Beach, CA.
- Uehling, Mark (2002), *Peeking at Big Pharma's IT Playbook*. Bio-IT World, 09/09/2002.
- Williams, Tim (2002), *A Version Control Kluge for SAS Programs – Using SAS*. Proceedings of the 27th Annual SAS Users Group International Conference, Orlando, FL.

Williams, Tim (2003), *A Programming Development Environment for SAS Programs*. Proceedings of the 28th Annual SAS Users Group International Conference, Orlando, FL.

ACKNOWLEDGMENTS

Special thanks to all of the life sciences organizations who provided their time, perspectives, and feedback to our team as we explored the concepts and issues discussed in this paper.

CONTACT INFORMATION

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

Pinnacle Solutions, Inc.
1 North Meridian Street, Suite 1010
Indianapolis, IN 46204
Work Phone: 317-423-9143
Web: www.psiconsultants.com

Curt Wehrley – curt.wehrley@psiconsultants.com

D.J. Penix – dj.penix@psiconsultants.com

Wing Lam – wing.lam@psiconsultants.com

SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration. Other brand and product names are trademarks of their respective companies.

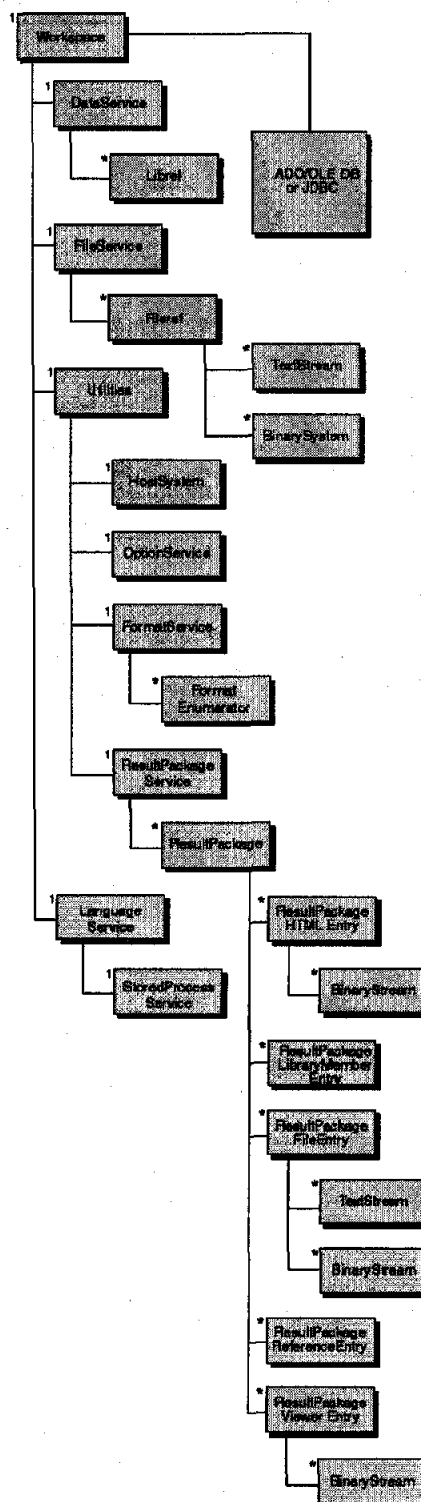
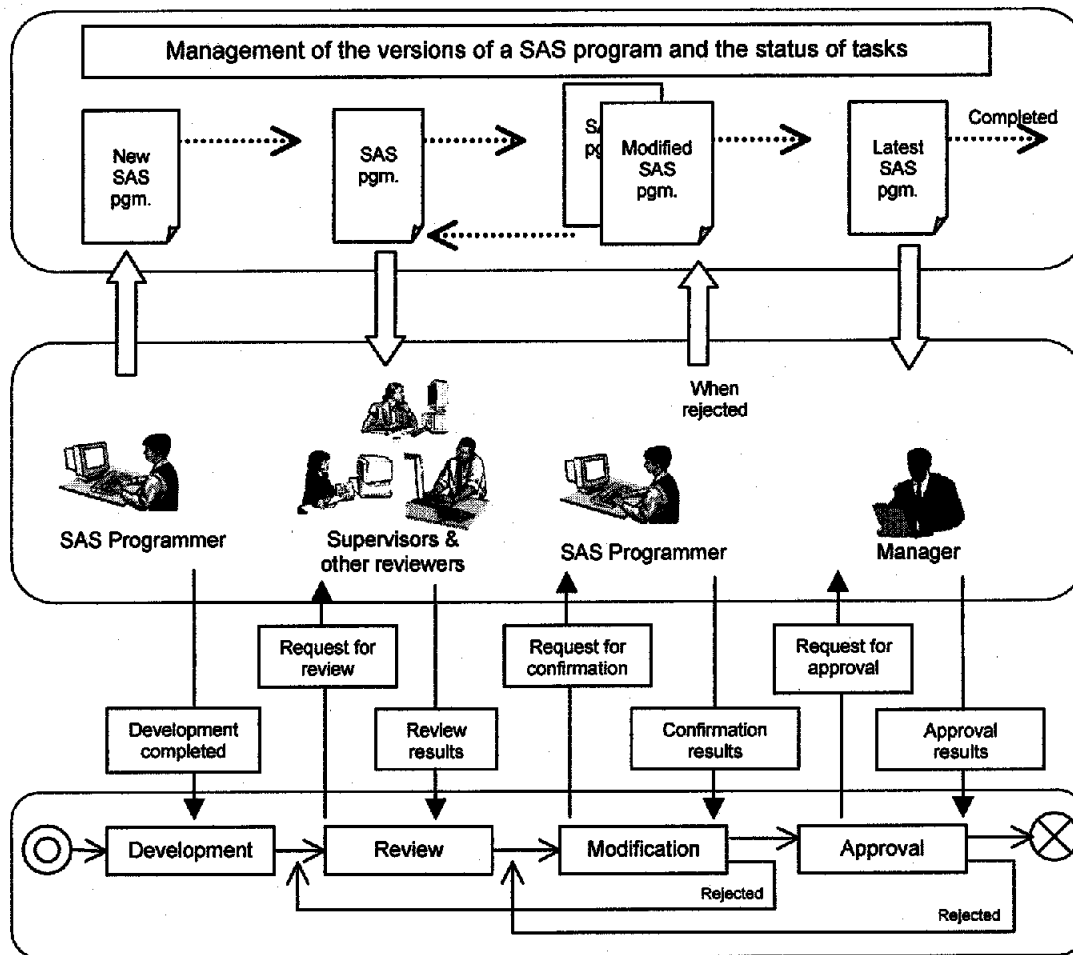


Figure 1: SAS Integrated Object Model (IOM)

Figure 2: SAS program lifecycle and workflow (pre-production stage*)



* - Figure describes a generic process for preparing a SAS object for production use. Does not include process for validating and managing SAS object once it is executed for actual data analysis.

Figure 3: Modero architecture

