ENTERPRISE DATA MANAGEMENT WITH STK DATA FEDERATE

Richard Page
Analytical Graphics, Inc. (AGI)

ABSTRACT

The STK Data Federate is a collaboration tool for managing and sharing STK data. Users can create and upload revisions of STK content, associated data files and documentation to the Data Federate repository, while automated processes support interoperability and data maintenance. It also provides tools for administrators to control which users have access to which files and expose custom services to facilitate searching rich metadata. The STK software suite integrates with the Data Federate server to allow users to easily store and retrieve content.

DATA FEDERATE FEATURES

- Document uploads, downloads and transfers
- Version control
- Access control and authentication
- XML and key-value metadata
- Email subscriptions for document updates
- Automated data backup and data processing
- Integrated with STK desktop
- Development kit for custom clients

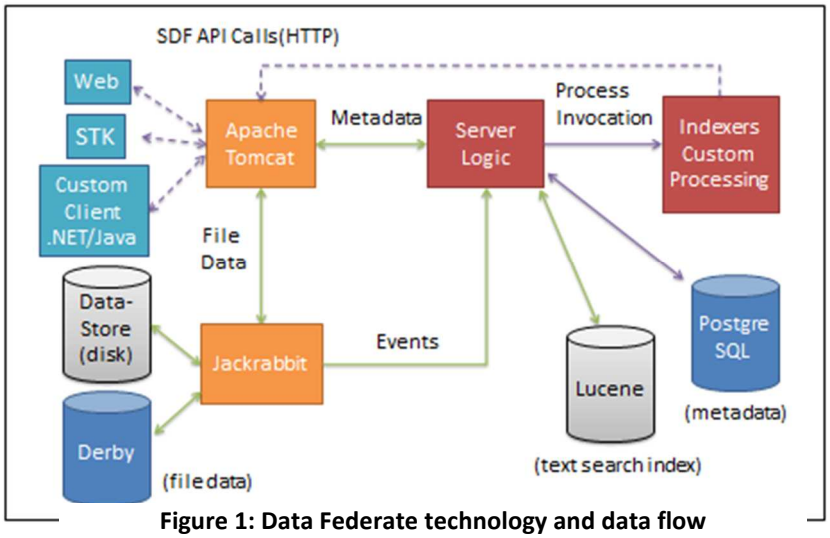

Figure 1: Data Federate technology and data flow

The Data Federate is a document repository built on Apache Tomcat which can host and manage file content from both STK and non-AGI products ranging from simple text documents to propriety binary files. For STK content, users can browse the repository on the Data Federate from within STK desktop based on their access privileges. They can thus easily access STK data, edit the content and publish the STK objects and scenarios back to the repository for other users. When collaborating, users can create revisions when uploading new files or replacing existing files. Plus, when searching for data from other users, the results of the search include the written descriptions and object names from STK, which are recorded automatically when uploading STK content. Outside of STK desktop, the HTML interface hosted by the Data Federate server provides a simple way to access and upload documents and their metadata

using a web browser. For more detailed content management, the “Analysis Explorer” tool (shown in Figure 2) manages files in the repository remotely. This includes the metadata “properties” associated with a given document, subscribing for notifications on changes to particular documents or folders, uploading and downloading files and, as an administrator, it allows modifications to the access permissions for any file or folder.

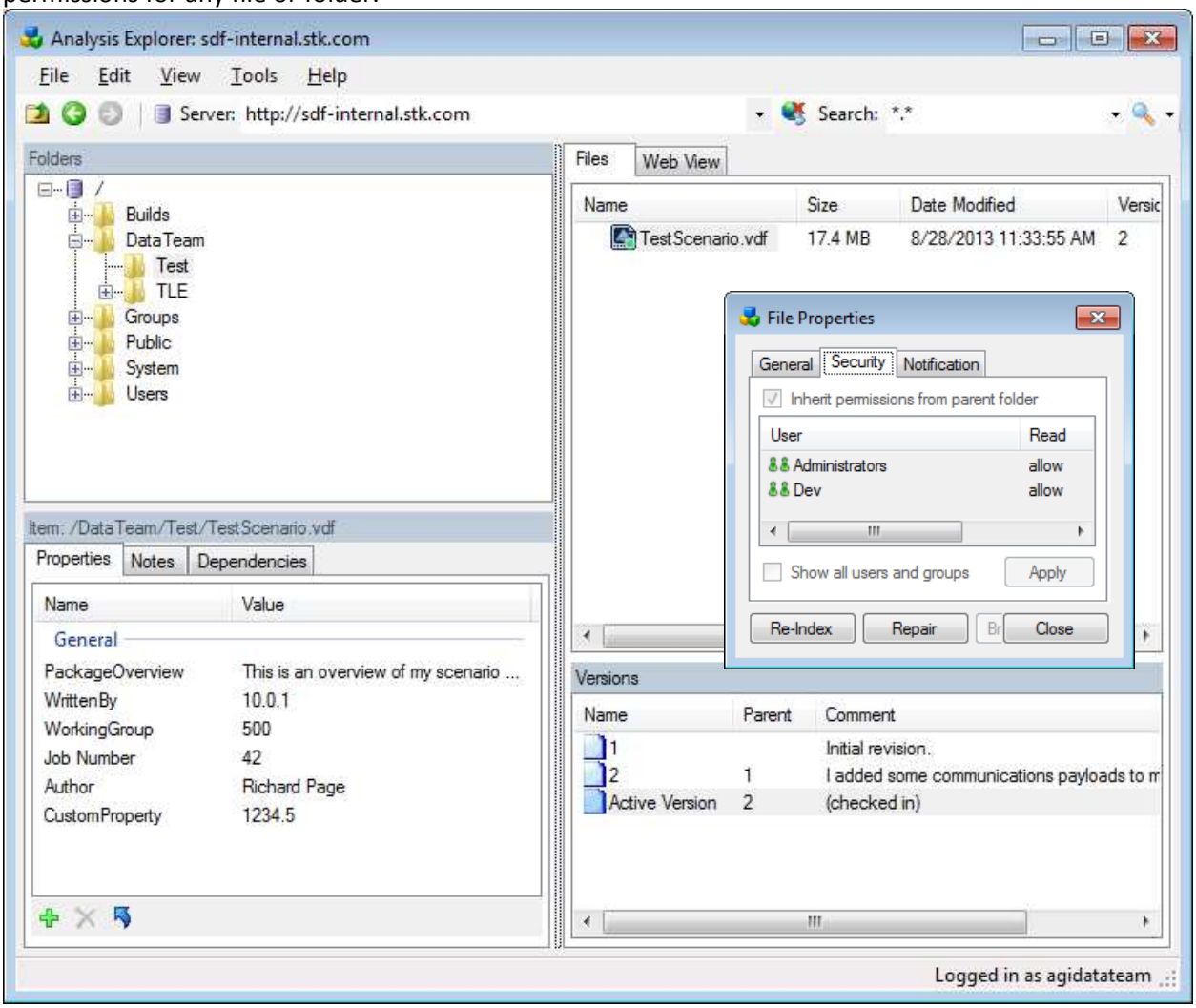

Figure 2: Analysis Explorer showing administration of property metadata, version control and access control

METADATA AND SEARCH

During collaboration, browsing to find specific archived data is often time consuming, and having the ability to search based on intelligent criteria is important. One of the key features of the Data Federate is the ability to associate metadata with documents in the repository to support search and retrieval. There are two kinds of metadata associated with Data Federate documents: XML and “properties.” The XML provides a way for STK or other custom clients to provide structured descriptions of obfuscated or non-textual data. When STK desktop uploads an object or scenario file, it compresses all of the child objects into a single package, uploads the package to the server and associates an XML file containing the names of all of the objects and their textual descriptions from inside STK desktop. That way, users can search based on content in addition to simple filenames. Furthermore, the Analysis Explorer (shown

in Figure 2) allows users and administrators to modify the "properties" on a given document. In this context, properties are just key-value pairs with a name and a textual or numeric value. Based on these, users can search for specific keywords, specific ranges of numeric values or object names. Other information, such as revision comments or additional notes added by an administrator are included in the general text search as well and show up as distinct entries in search results.

This metadata allows STK to provide a 'faceted' search based on complex criteria involving numeric ranges of orbital parameters, dates or enumerated parameters such as country of origin. One great example is the "Standard Object Database," which is a standard set of publicly available STK objects. Using a set of RESTful web services alongside the repository itself, the panel shown in Figure 3 is generated dynamically based on metadata provided by the Data Federate server. This allows updates to the search without requiring changes to the client software.

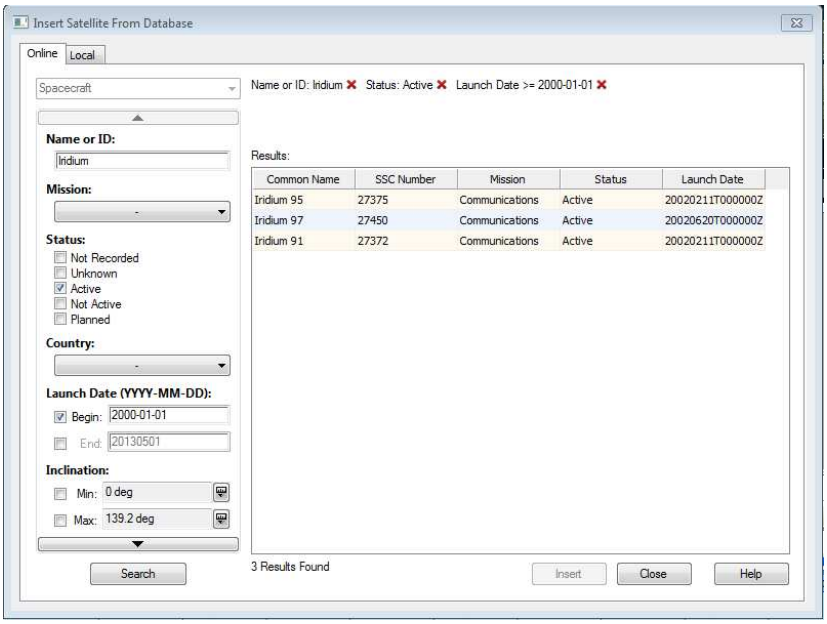

Figure 3: STK 10 Standard Object Database "faceted" search

FILE MANAGEMENT AND AUTOMATION

The Data Federate provides various ways of managing and automating data workflows. Analysis Explorer and STK desktop provide the simplest ways of managing content directly. However, working groups often require some customization to avoid repetitive maintenance tasks and manual data verification. To handle this need, the Data Federate can be customized either through custom client workflows or automated 'indexers' running in the background on the server.

Both custom clients and 'indexers' depend on the Data Federate development kits, which are available in both .NET and Java. This allows a custom client to perform upload and download operations, metadata management for both properties and XML, text and metadata searching, access control management and data transfers. Furthermore, using the development kit, combined with the STK development kits for STK UI Plugins, STK Engine or STK Components, enables the creation of custom workflows to support project-specific data management alongside existing STK features. Of course, Data Federate clients don't depend on STK and the server can be a simple way to support interoperability with other products.

One way to support interoperability and data verification is to provide customized "indexer" operations on the server. An indexer is essentially an executable compiled to integrate with the Data Federate server and run whenever a particular kind of document is modified in the repository. The interface provides an instance of the same client class available in the development kit along with information locating the triggering document. With that information, the indexer can perform automated tasks such as verifying data correctness, associating metadata for search based on data contents, generating

additional documents or initiating unrelated maintenance tasks on the server. In particular, indexers can be particularly useful to support interoperability by allowing conversions between file formats to occur automatically on the server. So, an STK desktop user may upload an STK object package, which then gets converted to one or more alternate file formats for use with other clients.

Lastly, tools are available to support transferring data from one server to another or to create archive copies of server data. In particular, an automatic backup system can be configured to create routine archives to ensure that if the data on the server ever gets corrupted, a recent backup can be restored after a fresh install. This backup contains the data content, metadata and user information.

USE CASE: COLLABORATING WITH THE STK DATA FEDERATE

Take the case of collaborating between three different roles on a single project:

- Adam is an aerospace engineer responsible for managing spacecraft orbit planning.
- Carol is a communications engineer responsible for analyzing performance.
- Sam is a systems engineer responsible for ensuring overall design success.

Adam is analyzing a change to his constellation of satellites. The satellites have varying degrees of maneuverability, and his goal is to propose updates to their orbits to improve the overall communications coverage on the ground. He uses STK Astrogator to plan several maneuvers to update the geometry of the constellation in a way that minimizes fuel consumption while hopefully improving coverage. Then, he uploads the updated STK satellites as revisions to the existing entries in the Data Federate with notes including the job number and descriptions of the changes.

Sometime later, Carol gets an email notification indicating there was an update to one or more of the satellites she was using. Now she needs to update her analysis. So, she downloads the Excel document she was using to model the proprietary receiver gains and loads the STK Engine application which allows her to plug her Excel models into the STK Communications analysis. A simple search for the job number pulls up the revision list for the new satellites and loads the latest version into the scenario. She can now evaluate the performance of the communications payloads to determine whether there are any changes necessary to the antenna pointing, frequencies or gains on each of the spacecraft. In addition, one of the spacecraft may be a new member of the constellation requiring her to attach a brand new set of STK Communications objects to the STK object package. After updating the communications payloads and running her analysis, she now uploads new revisions to the STK satellite objects (along with updates to her Excel document), with comments detailing her changes.

Meanwhile, Sam has created an “indexer” which he deployed on the server which starts a verification process when all of the design objects for a given design cycle have been updated and marked “completed.” It also intelligently checks for any discrepancies between revisions and sends an automated message to the engineer who uploaded the revision in question indicating there was an error in the data that needs to be addressed before the design cycle continues. When all entries in cycle are complete and tested, it automatically tags them as verified and proceeds to generate STK scenarios to run a trade study based on a prescribed set of parameters (launch dates, planned outages, external communications interference, etc.). Once the trade study is done, the new scenarios are uploaded to a separate folder along with documents detailing their results. Sam then gets an email indicating that a new trade study has results ready for review and he can go talk to Adam and Carol about the results.

Lastly, when the design cycle is complete and a new design is published to a production area, a separate server-side indexer processes the new STK scenario data to produce results available for visualization online and generate a detailed report of the changes since the last published design and the performance of the new system.

Whether a test suite is running in the background, an engineer needs to share models and analysis or a manager needs an easy way to understand progress on a design, the STK Data Federate provides a simple way to bring an enterprise together over a common set of data. To learn more about the STK Data Federate, call 1.610.981.8000 or write to info@agi.com.