

XML - James Webb Space Telescope Database Issues, Lessons, and Status

Ryan Detter
NASA/ Goddard Space Flight
Center
Greenbelt, MD 20771
301-286-9283
Ryan.D.Detter@nasa.gov

Michael Mooney
AURA/ STScI
JHU
Baltimore, MD
410-338-4508
mooney@stsci.edu

Curtis C. Fatig
Orbital Sciences Corporation
NASA/Goddard Space Flight
Center
Greenbelt, MD 20771
curtis.fatig@gsfc.nasa.gov

Abstract – This paper will present the current concept using eXtensible Markup Language (XML) as the underlying structure for the James Webb Space Telescope (JWST) database. The purpose of using XML is to provide a JWST database, independent of any portion of the ground system, yet still compatible with the various systems using a variety of different structures. The testing of the JWST Flight Software (FSW) started in 2002, yet the launch is scheduled for 2011 with a planned 5-year mission and a 5-year follow on option. The initial database and ground system elements, including the commands, telemetry, and ground system tools will be used for 19 years, plus post mission activities.

During the Integration and Test (I&T) phases of the JWST development, 24 distinct laboratories, each geographically dispersed, will have local database tools with an XML database. Each of these laboratories database tools will be used for the exporting and importing of data both locally and to a central database system, inputting data to the database certification process, and providing various reports. A centralized certified database repository will be maintained by the Space Telescope Science Institute (STScI), in Baltimore, Maryland, USA.

One of the challenges for the database is to be flexible enough to allow for the upgrade, addition or changing of individual items without effecting the entire ground system. Also, using XML should allow for the altering of the import and export formats needed by the various elements, tracking the verification/validation of each database item, allow many organizations to provide database inputs, and the merging of the many existing database processes into one central database structure throughout the JWST program.

Many National Aeronautics and Space Administration (NASA) projects have attempted to take advantage of open source and commercial technology. Often this causes a greater reliance on the use of Commercial-Off-The-Shelf (COTS), which is often limiting. In our review of the database requirements and the COTS software available, only very expensive COTS software will meet 90% of requirements. Even with the high projected initial cost of COTS, the development and support for custom code over the 19-year mission period was forecasted to be higher than the total licensing costs. A group did look at reusing existing database tools and formats. If the JWST database was already in a mature state, the reuse made sense, but with the database still needing to handling the addition of different types of command and telemetry structures, defining new spacecraft systems, accept input and export to systems which has not been defined yet, XML provided the flexibility desired. It remains to be determined whether the XML database will reduce the over all cost for the JWST mission.

TABLE OF CONTENTS

1. INTRODUCTION
2. THE ISSUE
3. WHAT IS XML?
4. GETTING OVER THE HUMP
5. COMMERCIAL OFF THE SHELF SOFTWARE
6. SUMMARY
7. REFERENCES
8. BIOGRAPHIES

1. INTRODUCTION

JWST is a large aperture infrared space telescope with a 10-year mission designated to succeed the Hubble Space Telescope (HST). JWST will continue the NASA tradition of advancing breakthroughs in our understanding of the origins of the earliest stars, galaxies, and the very elements that are the foundations of Life¹. Current launch date for JWST is 2011. Our goal is to substantially increase the amount science data available for downlink, with

IEEE paper 11301, Version 3 12/6/2003

0-7803-8155-6/04/\$17.00©2004 IEEE

guaranteed data delivery, at less cost when compared with HST.

The JWST team includes several partners in multiple locations: (1) Project management located at Goddard Space Flight Center (GSFC), (2) Observatory Prime Contactor, Northrop-Grumman Space Technologies (NGST), located in Redondo Beach California, (3) Integrated Instrument Module (ISIM) located at GSFC, (4) Near Infra Red Camera (NIRCam) instrument (University of Arizona and Lockheed Martin), (5) Near Infrared Multi-Object Spectrometer (NIRSpec) built in Europe, (6) Mid Infrared Instrument (MIRI) build jointly by US and European team, (7) Flight Guidance System (FGS) built in Canada (8) Deep Space Network, and (9) Science and Operations Center located in Space Telescope Science Institute (STScI) located in Baltimore MD.

For the JWST Project XML has been selected as the exchange format between the various systems needing database information. The first database delivery using XML was delivered by GSFC to the Telemetry and Command system and the Prime spacecraft contractor on September 30, 2003. Due to current contracts and

personnel working on the database had history from previous GSFC missions, and the past had shown the shorter missions generally did not experience any issues with obsolescence of the database and those systems needing database information. The longer term programs, such as HST, have not only experienced database software applications that are no longer supported, but also the changing out of systems needing the database information. The changing out of systems over time and the applications that become obsolete are generally an unplanned cost that must be absorbed. To switch to another application may sound easy, but for operational systems this involves at a minimum selection, testing, verification, training, and deployment of any new applications. These additional steps increase the total cost and as with most NASA missions, cost must be a consideration.

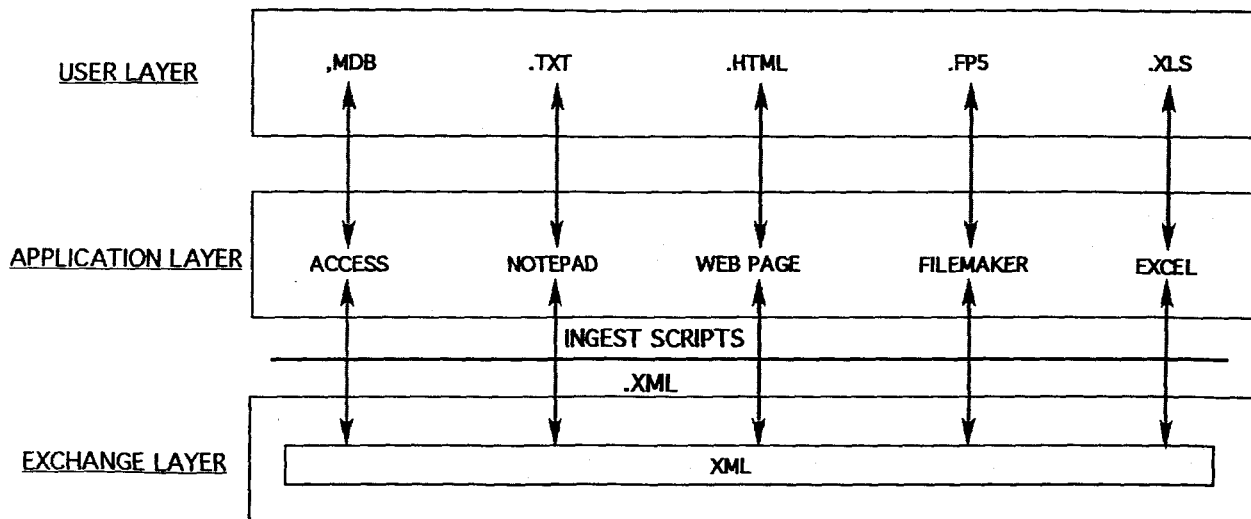


Figure 1 - Three layers of XML

deliverables to the JWST Project, the GSFC Common Command and Telemetry System group will be building and archiving JWST databases until the STScI Project Reference Database (PRD) Management System (PRDMS) is delivered

2. THE ISSUE

The JWST has been exploring the database structure and tools for approximately two years. During the first year no prime contractor or ground system was chosen. It became apparent that whatever was selected, a goal of usability from 2003 until at least 2021 is highly desirable. Many of the

3. WHAT IS XML ?

This may sound like an easy topic, but as we have learned in the various presentations, discussion forums, and status meetings, persons have been conditioned to associate an extension with an application. XML is considered open source and is a means to associate information.

XML is generally never seen by end-users. The various applications, with their Graphical User Interfaces (GUI) and file extensions, are what the end-user interfaces with. As can be seen in figure 1, the database group divided the database elements into three layers.

The purpose of dividing up the database elements was to convey the relationship of XML to the traditional user and applications layers.

The User layer is the nominal file extension attached to files within most operating systems. The file extension assists the user, and the operating system, in determining what application to use for opening, viewing, editing, and saving.

The Application layer is the application that will accept the XML information from the database and allow the user to interact with it using the User layer file extensions.

Structurally, the various applications exchange information and interface with the users to build a database from the XML exchange layer. Figure 2 shows an overview of the database build process.

The Exchange layer is the XML tagged information in the database. Below is an example of the exchange format for the JWST CCSDS packet level information.

```
<PktUnit>
  <PktAPID></PktAPID>
  <PktIdentifier></PktIdentifier>
  <PktTlmLength></PktTlmLength>
  <PktDescriptiveInfo>
    <PktPedigree></PktPedigree>
    <PktGroundBuildID></PktGroundBuildID>
    <PktFlightBuildID></PktFlightBuildID>
    <PktDescription></PktDescription>
    <PktDescriptionURL></PktDescriptionURL>
    <PktLongDescription></PktLongDescription>
    <PktSource></PktSource>
    <SlotNumber></SlotNumber>
  </PktDescriptiveInfo>
</PktUnit>2
```

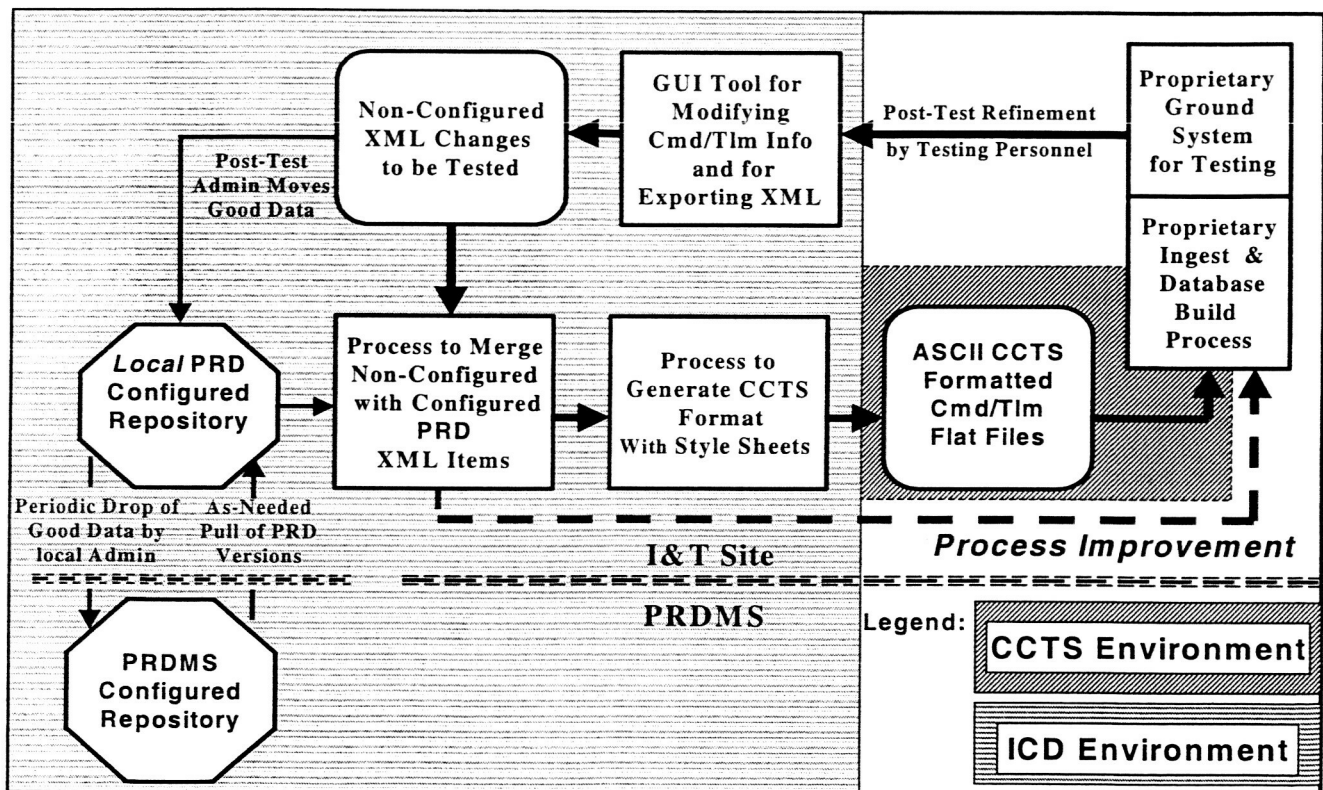


Figure 2 – Overview of STScI PRD Applications and Interfaces

Upon completion of the database build, the local I&T site will use the Project Reference Database Management System (PRDMS) to manipulate the database as needed. Some of the functionalities of this system include:

- Database viewing, modification, creation
- Real-Time, automated database validation
- Bulk database loads from central PRD
- Database versioning and cross referencing

A typical example of use of the PRDMS toolset at a local lab can be seen in figure 3.

As can be seen, each element needed has a unique identification associated with it called a TAG. For the JWST database each command and telemetry mnemonic will be a separate container. For example if there are 200 commands in the database, then 200 containers will be delivered in a command folder.

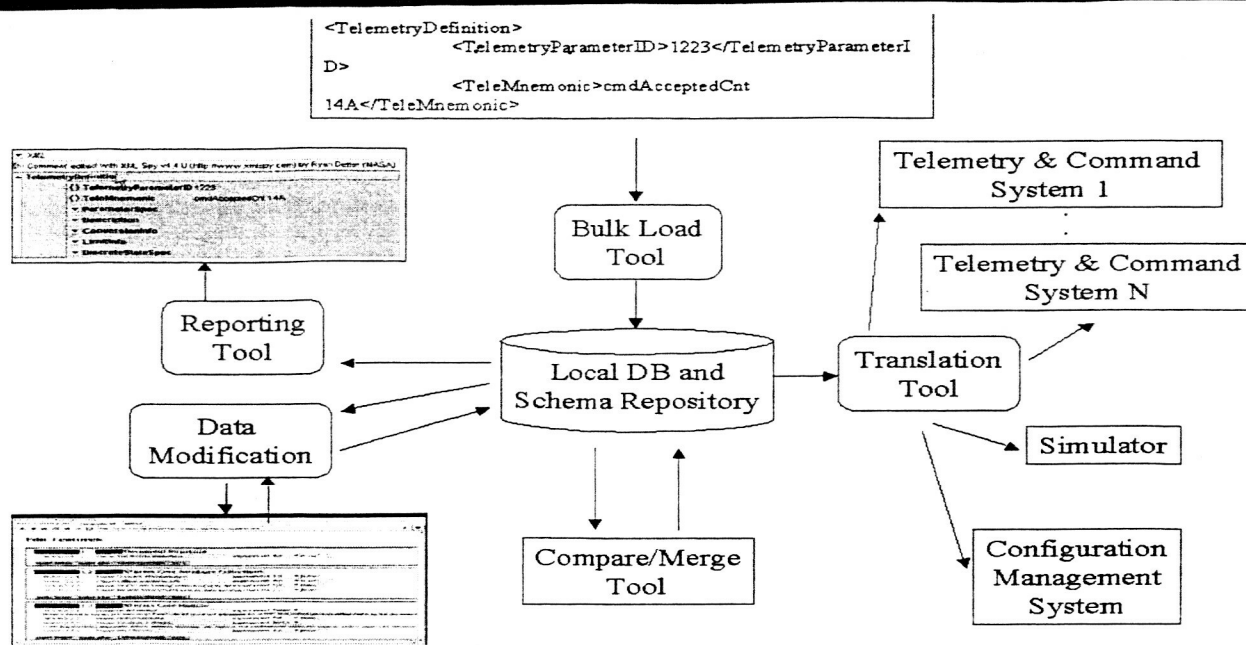


Figure 3 – Overview of STScI PRDMS Use-Case

4. GETTING OVER THE HUMP

As part of the effort to explain the entire database process and its operation in the Integration and Test (I&T) and real-time operational environments, the STScI produced a series of documents early in the mission phase. The JWSTProject Spacecraft Requirements review (SRR) is currently scheduled for 1st quarter 2004, yet the database is needed 6 months prior to that to support the Flight Software (FSW) development. As can be seen from figure 4 below, the FSW is only one of many groups that will be exchanging information with the central STScI PRD.

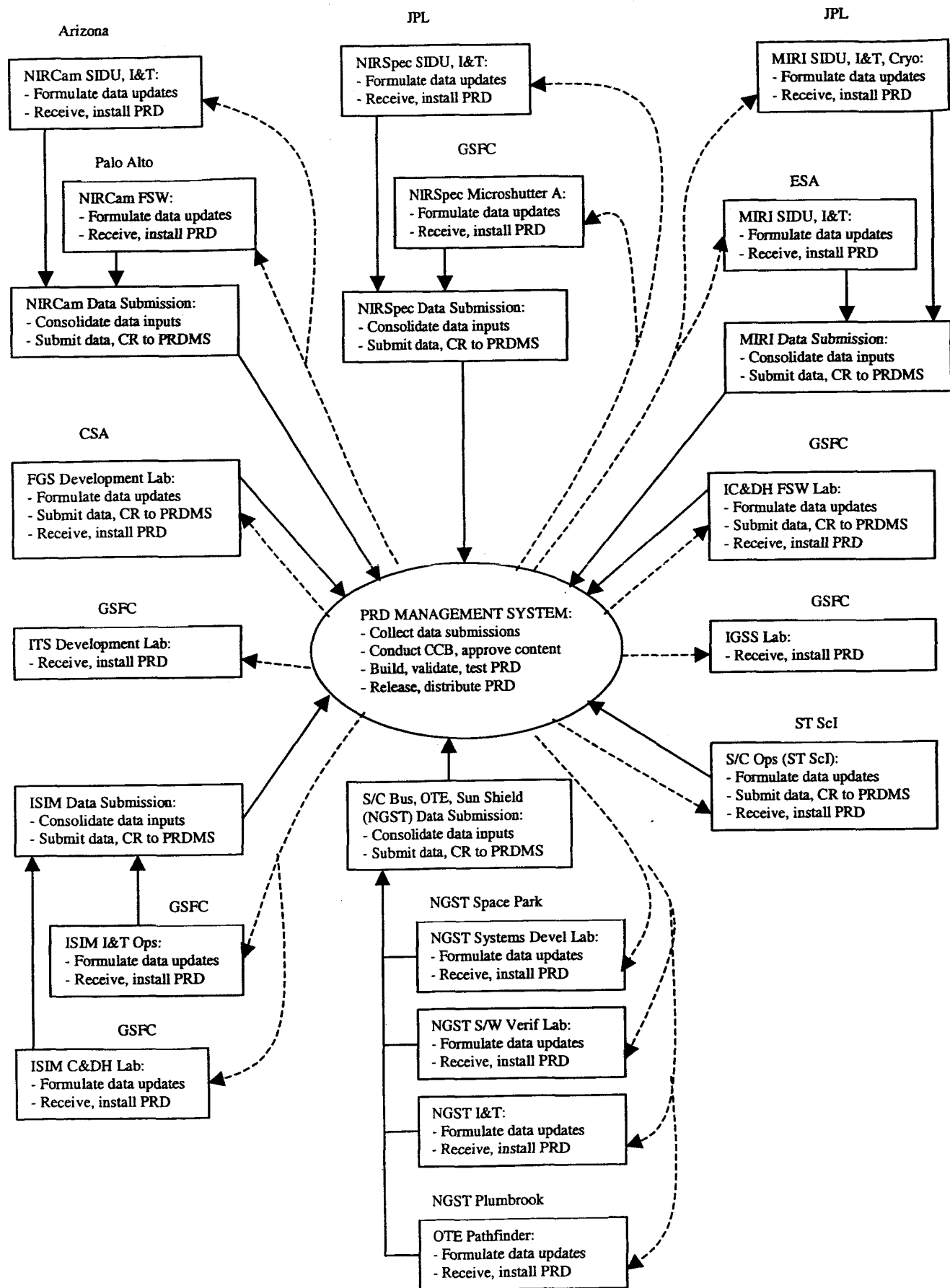
Most database use proprietary formats and structures. The JWST is to be shared across many systems and some of those systems have not been developed or selected at this time. The STScI PRD will contain all the information needed to operate the JWST and its various systems. The PRD includes, but is not limited to, the following data types³:

- Commands
- Commanding Scripts
- Activity Descriptions
- Observation Plan Interface Definitions
- Telemetry Definitions
- Engineering Unit Conversions
- Event Messages
- Display Page Definitions
- Aperture Definitions
- Spacecraft Characteristics (e.g. solar array dimensions)
- Constraints and Restrictions (e.g. attitude restrictions)
- PRD Environmental Specifications

All the various data types will have a XML container associated with it. Within each XML container will be the individual TAG'ged elements. These elements will be used by the other system, through an ingest script, to import the database.

One major task that had to be dealt with early on was coming up with a correct XML format (i.e. what elements would be present, what hierarchy they would be in, and what constraints would be on them) that could successfully fulfill the data requirements of the various consumers of data. This was handled in an evolutionary integration process with the various consumers; starting with the PRD, then including the Primary Contractor, the ground system, and eventually the science instruments and various simulators. This collaboration allowed for us to integrate unforeseen functionality both on our end as well as on the consumers end in the XML format definition. (Refer to Appendix A for the final XML Schemas)

The effort to build a database process, formats, and tools prior to the requirements being defined caused many issues to be discussed very early in the spacecraft development process. This has lead to changes occurring which in a normal development would not happen. Besides the frustration and the amount of time needed to make the changes, in the 'big picture' over the life of JWST this will be remembered as a minor inconvenience.



5. COMMERCIAL OFF THE SHELF SOFTWARE

This section discusses the products that were evaluated and used by the CCTS personnel. The JWST Project, early in the database design process, realized XML was the desired exchange format. Once XML was selected for the data exchange format, it was easy to determine that COTS products were the most cost effective means to implement the PRDMS for the prototype system and operations.

XML allows for easier integration with COTS than in the past because it is an international recognized standard that is implemented throughout industry. This not only allows for a robust infrastructure of tools available for selection, but also allows for standard integration of your XML formats into those applications.

COTS tools for using, manipulating, and translating XML are wide and varied. During the various COTS product evaluations this actually presented a more of a problem, too many choices. In general when selecting COTS products personnel preferences often cloud the judgment of the evaluators, so getting a fair scoring is often difficult. Also due to money and time constraints, the COTS evaluation was limited in scope. It has been determined that the XMLspy, IBM, and TIBCO product lines have the leading COTS products currently available for your XML needs. In addition to viewing, importing and manipulating the XML data structures, there are many COTS tools for displaying the data. Our team plans on using the standard web interfaces for the graphical interface, but with sophisticated tools to validate the end-user data at input. At this time the final selection has not been made.

There is much evidence that a well thought out, well implemented COTS solution will improve the speed and reduce expenses of our software development projects. There is further evidence that these benefits are often not as great as most of us would like them to be. It is certainly tempting to think, when planning a project, that the reports

and charts will be a breeze because they will all be implemented with COTS products. Care should be taken when planning a project to consider all of the activities that should be performed and to schedule time and resources to complete these activities adequately. It is also important to look at the long term issues associated with the solutions you select.⁴

6. SUMMARY

The database is only one piece of the entire JWST satellite project puzzle. As has been pointed out, with long-term mission, XML provides many benefits.

- Long-term support
- Non-Proprietary format
- Easily convertible to existing applications

Understanding that XML is an exchange format and not a file extension associated with a particular application is key. Persons are concerned about reading and writing XML, as is generally done between file extensions and applications. IN our system XML is a data exchange format between computer systems, persons will view the data using a web interface that will present the data in a user friendly manner. Stressing that XML is an open standard exchange format and what this means has been one of the challenges. More time has been spent on the selection of XML than on any of the COTS tools.

COTS tools for XML are very wide and varied. To cobble together various low cost and free tools can be more expensive in the long run than choosing a more expensive tool that meets all the needs. Often it is not the initial price of COTS that is the main cost, but the price of upgrades, licenses, testing, integration, and deployment.

XML is an excellent exchange format and would recommend its use for any database structure.

7. REFERENCES

- [1.] Mather, J., 'NGST Science Goals',
<http://www.ngst.nasa.gov>.
- [2.] JWST Project Prototype Database Interface Control
Document – Draft September 2003
- [3.] JWST Project Reference Database Management
System Operations Concept – Draft July 2003
- [4.] Minkiewicz. A., "The Real Cost of COTS", Paper
presented at the IEEE Aerospace Conference 2001,
Bigsky, MT, March 2001

8. BIOGRAPHIES

Ryan Detter is a Real-Time Software Engineer for the ISIM Ground Support System for the James Webb Space Telescope Mission. He is currently working on the development of the XML ground system tools as well as the Interface Control Document that will be used at the various distributed data sites. He also works on the test team for the JWST electrical bus, i.e., Spacewire. He received a B.S. in Electrical Engineering from Ohio University in 2002.

Mike Mooney is a Software Systems Engineer at the Space Telescope Science Institute in Baltimore, Maryland. His current assignment is JWST Flight Operations System Engineer. Mr. Mooney has 25+ years of experience in the areas of satellite command, control, and communication systems; naval combat systems engagement simulation operations; orbital and sub-orbital launch operations; launch range and flight safety systems. Prior to his current assignment, he provided flight software development, ground systems engineering and project management to various NASA missions including ADEOS, ISTP, GOES, NPP and TIMED and DoD missions.

Curtis Fatig is the lead Ground System Syetem Engineer for the JWST Common Command and Telemetry System. Prior to working on JWST, he served as the lead Hubble Space Telescope Servicing Mission Test and Integration Engineer. His team of engineers set up remote Payload Operations Centers, test the entire ground and space software and communication links used for each HST servicing mission, and test NASA institutional upgrades affecting HST. He also supports long term development of new mission ground systems. He received NASA's Spaceflight Awareness Award, Public Service Medal, and many Achievement Awards for these efforts from GSFC, KSC and JSC. He received a B.S. from Salisbury State College in 1978 and a M.Ed. from Rollins College in 1982.