

# SOAS: Developing Multi-Mission Architecture for Astronomical Observatory Scheduling

Joe Chavez

California Institute of Technology  
1200 E. California Blvd, M/S 100-22  
M/S 100-22  
626-397-7166

jchavez@ipac.caltech.edu

## ABSTRACT

The Space Infrared Telescope Facility (SIRTF) is the fourth and final element in NASA's family of "Great Observatories" and is set for a July 2002 launch. SIRTF will provide infrared imaging and spectroscopy in the 3 to 180-micron wavelength range. The Science Operations Application Server (SOAS) supports the SIRTF Planning Observations Tool (SPOT) and Science Operations System (SOS). The SOAS will provide Internet enabled web services that include proposal submission, astronomical catalogs, automated software update, object persistence, and instrument simulation[1].

## Categories and Subject Descriptors

- Components and frameworks,
- Object databases and persistence,
- Patterns and software architecture,
- Java tools, compilers and technologies,
- UML tools, technologies, and methodologies,
- XML tools, technologies, and methodologies,
- Objects and the web,

## General Terms

Performance, Design, Economics, Reliability, Security.

## Keywords

Java, XML, Objects and the Web, UML, Web Services.

## 1. INTRODUCTION

The Science Operations Application Server (SOAS) supports the proposal creation, editing and submission activities the SIRTF Science Operations System. The architecture is an n-tier, high-

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

Conference '00, Month 1-2, 2000, City, State.

Copyright 2000 ACM 1-58113-000-0/00/0000...\$5.00.

availability, scalable, transactional, service based design. The design and analysis of the software architecture is based on Objected Oriented Analysis and Design techniques and is expressed in the Unified Modeling Language (UML). The system is implemented using the Java Development Kit (JDK) and utilizes features from the Java 2 Enterprise Edition (J2EE). The SOAS software architecture is comprised of web services, Enterprise Java Beans (EJB), security and system monitoring components.

## 2. ARCHITECTURE

The SOAS is based on an n-tier architecture design[2] with client (SPOT), application server (SOAS), and data (DBMS) tiers, shown in UML Deployment Diagram in Figure 1[5]. The SOAS is comprised of the WebLogic Application Server, Java 2 Enterprise Edition SDK and the Solaris operating system. All software runs on Sun hardware. The communications protocol between client and server is HTTP/HTTPS. Use of this protocol allows any Java or browser client with Internet access to communicate with the SOAS. Back end services (Data and Visibility) may select the most appropriate protocol suitable for integration with the SOAS.

Applying OOA&D techniques on the high-level architecture reveals the set of SOAS components shown in Figure 1[4]. The logical components are grouped into physical tiers:

- Client Tier
- Application Server Tier
- Database Tier
- Computing Engine Tier

The Client Tier is composed of:

- SPOT – SIRTF Planning and Observation Tool
- IManager Interface
- Security Manager Proxy
- Program Manager Proxy
- SOAS Services

The Application Server Tier is composed of:

- Security Manager Service
- Security EJB
- Program Manager Service

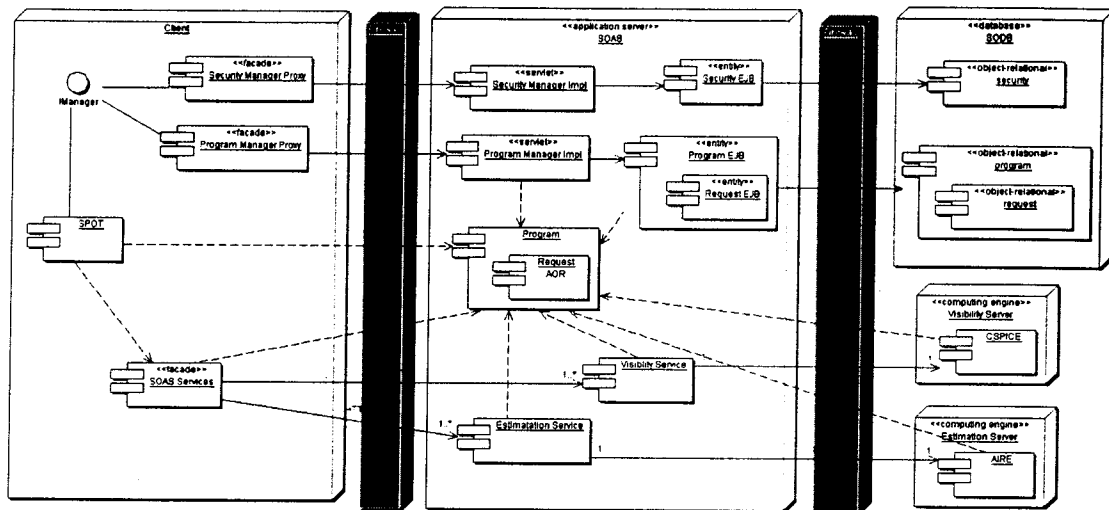


Figure 1 - SOAS Component/Deployment UML Diagram

- Program EJB
- Program Value Object
- Visibility Service
- Estimation Service

The Database Tier is composed of:

- Security Persistence
- Program Persistence

The Computing Engine Tier is composed of:

- Visibility Server
- Estimation Server

### 3. Frameworks

The Security Manager and Program Manager services are based on design frameworks that can be reused on subsequent missions.

#### 3.1 Persistence Framework

The Program Manager is built upon the Persistent Framework, which is primarily based on existing design patterns from the GoF[6] and Java 2 Enterprise Edition[3]. The framework defines the interfaces for Java object persistence from a remote client. The framework consists of a set Java interfaces for which classes are implemented to realize the interfaces. The framework provides both the remote client and server implementation for the services provided. The remote client implementation encapsulates all the protocol handling, data marshalling and security specifics. The server implementation encapsulates the persistence logic and handles persistence services such as creating, reading, updating and deleting object in the data store.

#### 3.2 Security Framework

The Security Manager is built upon the Security Framework, which is based on existing design patterns (referenced in 3.1 Persistence Framework). The Security Framework encapsulates the authentication and authorization of remote users of the system.

### 4. REFERENCES

- [1] Chavez, J. 2000, Multi-tier Internet Architecture with Java, UML and OOA & D in ASP Conf. Ser., Vol. 216, Astronomical Data Analysis Software and Systems IX, eds. N. Manset, C. Veillet, D. Crabtree (San Francisco: ASP), 75
- [2] Heinrichsen, I., Chavez, J., Hartley, B., Mei, Y., Potts, S., Roby, T., Turek, G., Valjavec, E., & Wu, X. 2000, The Scientific Uplink and User Support System for SIRTf in ASP Conf. Ser., Vol. 216, Astronomical Data Analysis Software and Systems IX, eds. N. Manset, C. Veillet, D. Crabtree (San Francisco: ASP), 319
- [3] Alur, D., Crupi, J., Malks, D., Core J2EE Patterns, Java 2 Platform, Enterprise Edition Series. Sun Microsystems Press, Upper Saddle River NJ, 2001
- [4] Booch, G., Jacobson, I., Rumbaugh, J., The Unified Software Development Process. Addison-Wesley, Reading MA, 1998
- [5] Booch, G., Rumbaugh, J., Jacobson, I., The Unified Modeling Language User Guide, Addison-Wesley, Reading MA, 1998
- [6] Gamma, E. et. al. Design Patterns, Addison-Wesley, Reading MA, 1995