
# Integrating RRS with Data Services in a Services Oriented Architecture

## Functional Requirements Document Presentation

June 1, 2004

APEX DIGITAL SYSTEMS

# Agenda

- Introduction
- Background: Apex's Software Architecture Engagement Summary for OHD
- Current activity: Requirements for a data service for RRS
- Perceived threats to migration flexibility
- Architectural considerations for a migration strategy
- Key migration elements
- Recommendation: Adapter-based design
- Requirements related to upgrading RRS
- Data services architecture components
- Conclusions & next steps

# Introduction

- OHD has been working with Apex on migration-related topics for some time.
- In 2003, OHD requested that Apex implement a proof-of-concept workflow management system that incorporates key services-oriented architecture features.
- In late 2003, OHD requested that Apex perform design activities to define a potential future software architecture based on a services-oriented architecture. Apex delivered this design to OHD in early January 2004.
- Recently, OHD requested that Apex define specific functional requirements for a data service, in the context of the NWSRFS pre-processor RRS. The data service is intended to show the specific capabilities of a services-oriented architecture vis-à-vis an actual NWSRFS component.

# Background: Apex's Software Architecture Engagement Summary for OHD

- The previous software architecture engagement described for OHD key process and technical themes relevant to defining a future software architecture.
- As a result of the engagement, Apex defined a possible high-level, services-oriented architecture that includes
  - Algorithm services
  - Display applications
  - Control services
  - Data services

# Current activity: Requirements for a data service for RRS

- OHD requested that Apex define clear requirements for design of the data service architecture recommended previously.
- OHD further requested that the requirements be defined in the context of modifying a NWSRFS pre-processor, RRS, to use the designed data service.
- Apex performed on-site and off-site requirements discovery.
- Based on the results of discovery related to RRS, Apex also evaluated important migration strategy issues that influence the requirements.

# Perceived threats to migration flexibility

- The discovery activities related to RRS revealed two important threats to migration flexibility:
  - RRS' loop control structures are closely related to the underlying database implementation.
  - RRS' science code has intimate knowledge of the underlying database implementation.
- Most likely, RRS was highly optimized for the limited availability of memory at the time RRS was created. Today, constraints focus more on network bandwidth.
- Migrating RRS to a data services architecture will be challenged by the tightly coupled loop control and the tightly coupled interface to the database.
- Any migration considerations must address ways to abstract the scientific code from the structure of underlying repositories.

# Architectural considerations for a migration strategy

- When modernizing or migrating current software components such as RRS to a new architecture, OHD will have several implementation options:
  - Minimal impact option: re-write only input/output functions, connecting them to newer data sources
  - Moderate impact option: define standard API for data exchange with data sources, implement an adapter-based architecture
  - Significant impact option: re-write entire application such as RRS, including underlying data sources.
- Any next steps related to modernizing current applications should focus on the moderate impact option, to ensure increased long-term code stability.

# Key migration elements

- The design should allow incremental implementation.
- The design should enable performance optimizations for specific data sources.
- The design should enable future changes to RRS data sources with minimal code changes to the core interfaces.
- The design should promote stability and correctness while changing data sources.
- The design should enable data sources to be tested in isolation, prior to integration with scientific code.

# Recommendation: Adapter-based design

- The adapter-based design separates science code from the implementation of underlying repositories.
- Adapters implement a standard API (application programming interface) for each scientific applications, while implementing repository-specific data access capabilities.
- Adapters function equally in a services-oriented architecture or in an architecture centered around local application execution.
- Adapters enable rapid exchange of repositories with minimal impact on the scientific application using the data provided by the repository, creating migration flexibility.

# Requirements related to upgrading RRS

- Apex developed high-level requirements for RRS' functionality
  - Managing stations and parameters
  - Managing observations
  - Managing time series
  - Managing record locks
- To implement a services-oriented architecture around RRS, the core scientific code does not require modifications.
- RRS currently accesses its data without any implementation abstraction – a basic abstraction layer will be required for modernization. We recommend using an adapter-based design.

# Data services architecture components

- In the previous engagement, Apex defined a key set of components for a services-oriented architecture. For the current engagement, Apex modified the originally recommended architecture to include the following components:
  - Science Application Interface (SAI)
  - Data Location Interrogator (DLI)
  - Data Service Adapter (DSA)
  - Data Service Adapter Factory (DSAF)
  - Data Access Controller (DAC)
  - Directory Service (DS)
  - Lock Management Service (LMS)
- The Functional Requirements Document outlines specific functional requirements each component must fulfill.

# Conclusions and next steps

- A services-oriented architecture, as previously defined, fulfills key requirements for improved science operations and research.
- An adapter-based design, implemented in the context of a services-oriented architecture, fulfills technical modernization requirements.
- To determine the suitability and performance characteristics of a services-oriented adapter technology, OHD should proceed to implement a proof-of-concept version.