COSMOS Demo

By the COSMOS Team

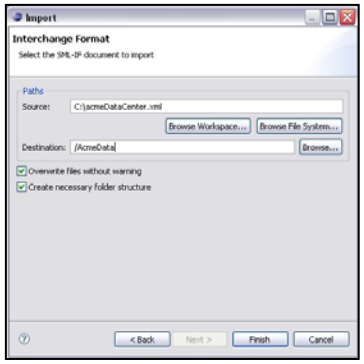
COSMOS Milestone 1 overview

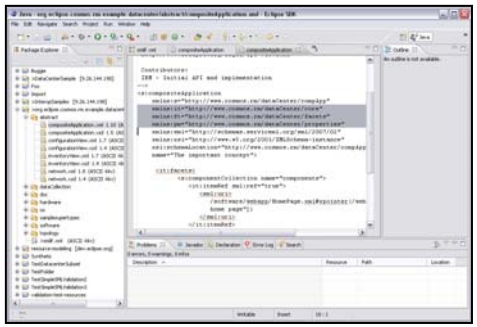
Milestone 1 (M1) shows an initial driver of a systems management framework based on open and emerging industry standards including the Service Modeling Language (SML), Web Services Distributed Management (WSDM), and Java Management Extensions (JMX)

An example application demonstrates the key features of the project including tools for the validation of SML and SML-IF documents, collection and persistence of management data based on event and statistical data, and simplistic reports for the visualization of the collected information.

Demo Overview
Working with SML & SML-IF documents

Systems Integrator



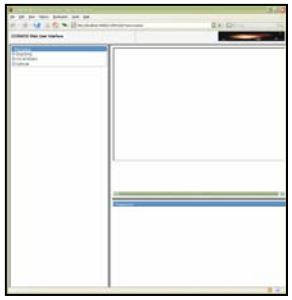


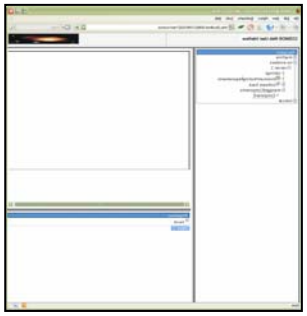
1. **The Systems Integrator receives an SML-IF document. This can be the result of an export from a configuration management database, another management system, or simply created by hand.**

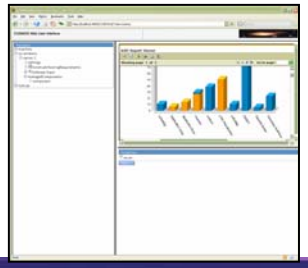
2. **Using the COSMOS SML support, the Systems Integrator imports the SML-IF document with the Eclipse workbench.**
   - **The SML-IF document is expanded into its constituent parts and stored on a file system repository**
3. **He would like to be able to associate management data contained in other databases**
   - **He then selects resources and adds the key set metadata that links into the management system**
   - **This is event data he got by importing log files from CBE (common base event) and statistical information collected using JMX**
4. **He then exports the resources to a new SML-IF document so it can be served by an application server**

Demo Overview:
Browse IT Resources

System Admin

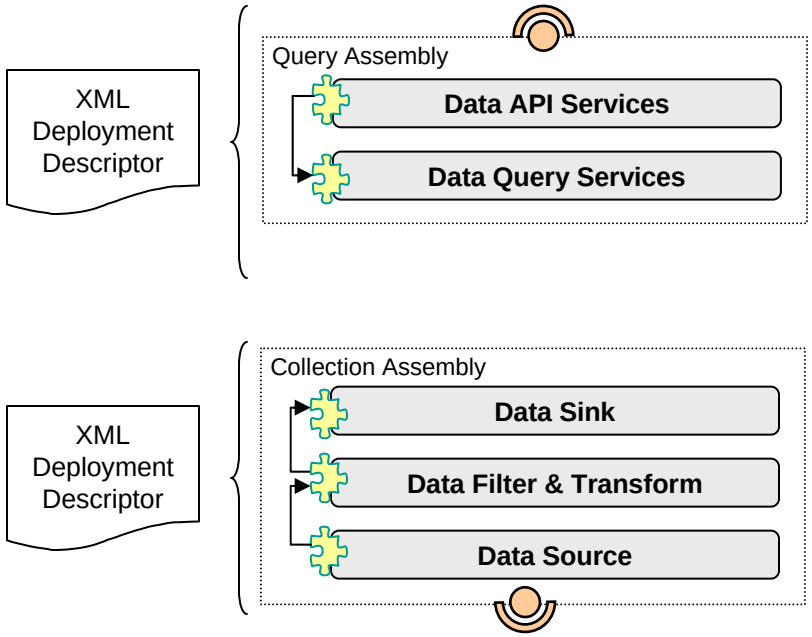






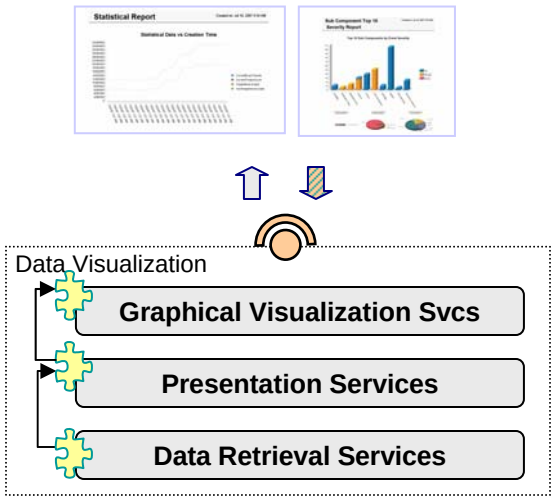
1. **The System Admin enters the starting URL of the COSMOS application.**
   - **A predefined navigation list of resource types are presented.**
2. **The System Admin wants to view information about the Windows servers present in their environment. She expands the "os.windows" node.**
   - **The Windows instances are listed.**
3. **She then expands out the "Server 1" instance.**
   - **This node contains four entries, "settings", "software stack", "minimum hosting requirements", and "managed elements"**
   - **Each of these entries corresponds to a "facet" in the definition of the resource.**
4. **Navigating to the "settings" and "minimum hosting requirements" nodes, the properties sheet is updated with the resource information.**
5. **Navigating to the "managed components" entry, the System Admin expands the keyset property.**
   - **This node represents a relationship between the Windows operating system and a component that is used for systems management. This component reports data in Common Base Event format.**
6. **The COSMOS application has a set of visualizations registered to handle Common Base Events. The user selects the "Top 10 Log Report" from the Report menu.**
   - **The report is displayed in the view above the properties sheet.**

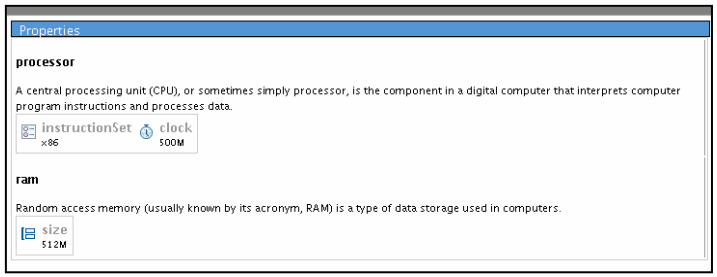
Key aspects of COSMOS Data Collection



- **The Data Collection component of COSMOS provides an extensible collection of components which may be configured into a set of Data Collection Assemblies and Query Assemblies.**
- **Both types of assemblies are described via a simple XML document—similar to a deployment descriptor. The COSMOS runtime processes this document and instantiates the physical manifestation of the assembly.**
- ***A key design element is for commercial adopters to extend example assemblies or provide their own to the COSMOS framework.***
- **Query Assemblies facilitate integration of management data by providing a common services abstraction layer on top of a data store. Web services based interfaces provide standard access mechanisms to the data.**
  - **COSMOS provides an example data store, but it is expected that adopters with existing databases will simply wrap their current implementations**
- **Collection assemblies receive their input through standards based interfaces, e.g. JMX and WSDM (CBE is also leveraged as it is considered a precursor to the WEF standard)**
- **Since the Data Sink and Data Query Service are pluggable components, any type of data store may be leveraged, including existing management databases.**
  - **COSMOS provides a simplistic data store to demonstrate how the framework is used**

Key aspects of COSMOS Data Reporting

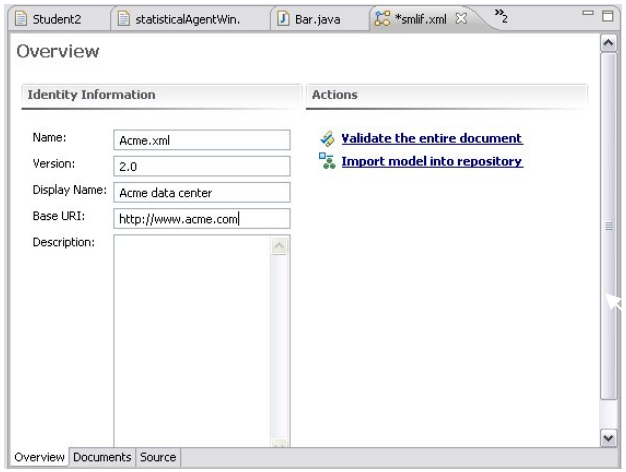


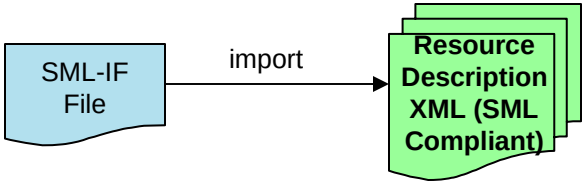


Minimum Hosting Requirements

- **COSMOS presents an example application to demonstrate the different parts of the framework.**
- **Browsing is accomplished by traversing a pre-defined resource graph representing the "top level" nodes.**
- **Resource information is obtained by importing an SML-IF document. The defined top-level nodes are matched up with entries in the SML documents and information is presented to the user**
- **Because SML based descriptions of resources may be composed of "facets", COSMOS has the ability to register visualizations based on a facet.**
  - **An example visualization for "Minimum Hosting Requirements" is shown.**
  - **The COSMOS framework provides the ability to define visualization conventions for resources with standardized facets.**
- **BIRT based reports are another example of a visualization based upon a resource's facet.**
  - **COSMOS provides a small collection of example reports to demonstrate this framework.**

Key aspects of COSMOS SML support





- **SML-IF and SML validation**
- **SML-IF editing**
- **SML-IF import and export**
- **File based repository with service based API**
- **Example SML-based representation of a data center, including management infrastructure**

Summary

- COSMOS provides a systems management framework that can be deployed as-is, or extended to integrate with various systems and data stores
- COSMOS embraces industry standard specifications to promote cross-vendor integration and easy adoption
- COSMOS provides an example end-to-end scenario that illustrates one of its typical use cases