Community Systems Management Open Source "COSMOS" Creation Review

DRAFT v0.5
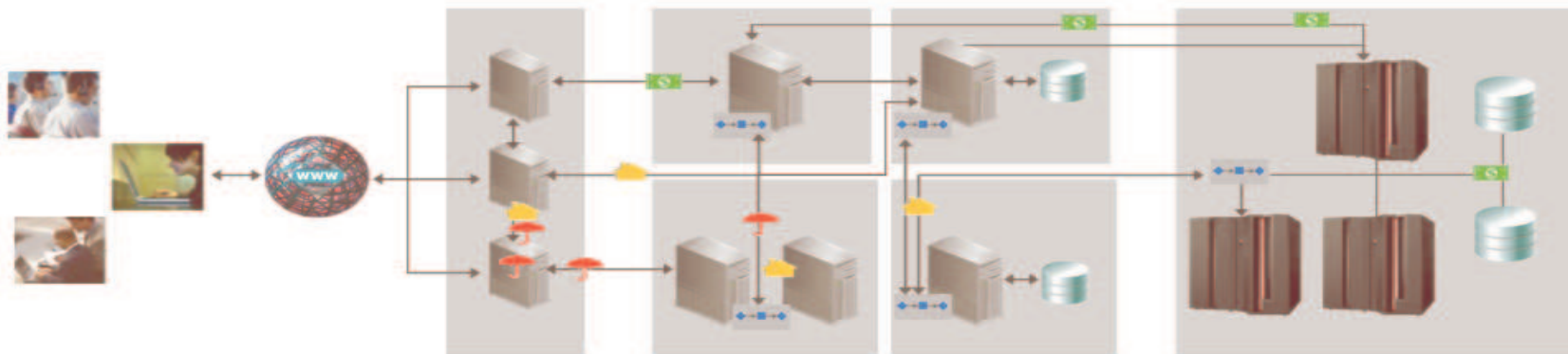
Put together by the COSMOS community
http://wiki.eclipse.org/index.php/COSMOS

Agenda:

- Brief summary of the project

- Initial implementation focus

- Current project participants

- Comments about the community response to the proposal

- Confirmation that the project members have read and understand the Eclipse Development process and these guidelines

Challenges of IT Systems Management

Today's applications offer flexibility for business but introduce management challenges …

"How do we coordinate problem resolution across all parts of the organization?"

"How can we minimize the disparate information we collect?"

"How can we instrument our systems?"

"How can we share resource descriptions in a consistent manner?"

"What industry standards can be applied to help solve the problems?"

"I know something's wrong, but where?"

COmmunity Systems Management Open Source*

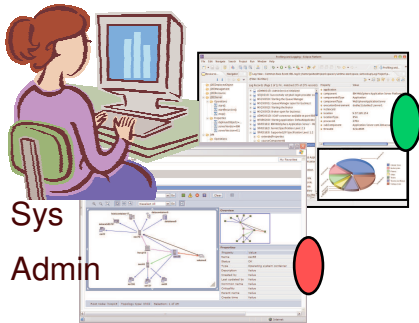- **COSMOS project proposal submitted to Eclipse**
  - Built upon the Eclipse Platform, the COSMOS project intends to develop generic, extensible, standards-based components for a tools platform upon which software developers can create specialized, differentiated, and inter-operable offerings of tools for system management

    - http://www.eclipse.org/proposals/cosmos/
    - http://wiki.eclipse.org/index.php/COSMOS

  - The framework will exploit standards-based services for monitoring, and modeling of enterprise resources, applications, and workloads
    - COSMOS will facilitate the next evolutionary move for systems management
    - COSMOS will define the common assets needed in a platform for vendors to derive immediate benefit and in turn provide higher value in their own offerings
    - COSMOS will facilitate a migration to a common and open standards driven set of components
    - COSMOS will define tooling for modeling resources based on SML standard

  - In accordance with Eclipse process the COSMOS project will incubate following the guidance and supervision of the Eclipse Technology PMC
    - Expectation is COSMOS will graduate to a top level Eclipse project in March 2007 along with the GA of release 1.

*COSMOS is a proposed name. The community will determine the project name.

COSMOS: Key Components

- **Resource Modeling**

- **Build to Manage (BtM) a.k.a Enabling Management**

- **Monitoring**

    - Data collection

    - Reporting

Current pain points

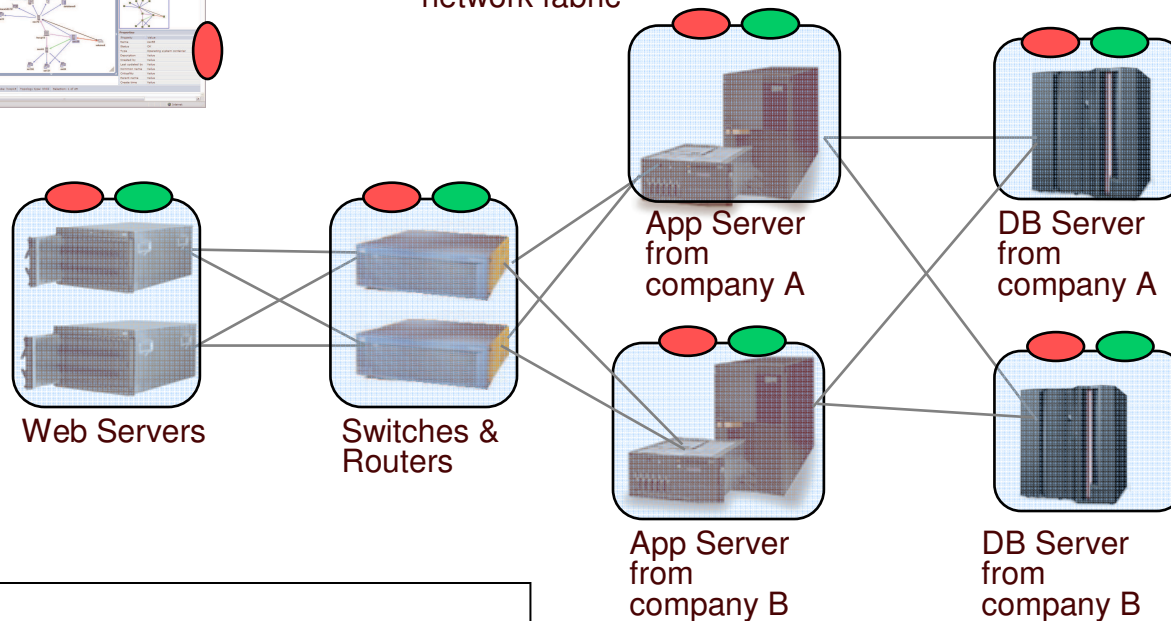Each admin tool has a distinct way of representing the resources it can manage

Limited interoperability b/t tools (even those from the same company)

Limited co-existence of management tools at runtime

Management tools typically targeted at a specific resource domain e.g. network fabric

Customer Pain Points

▪End-to-end problem determination is awkward and difficult

▪Response to change in requirements is slow

▪Creating higher value analytics difficult b/c of inconsistencies in the way resource information is expressed

▪Inefficiencies in management costs displace development work

Sys

Admin

Often, this translates into custom instrumentation provided by management vendors.

This results in subtle, but important semantic differences in the way a resource is managed

App Server from company A

DB Server from company A

Web Servers

Switches & Routers

App Server from company B

DB Server from company B

Vendor Specific Management APIs

Current pain points
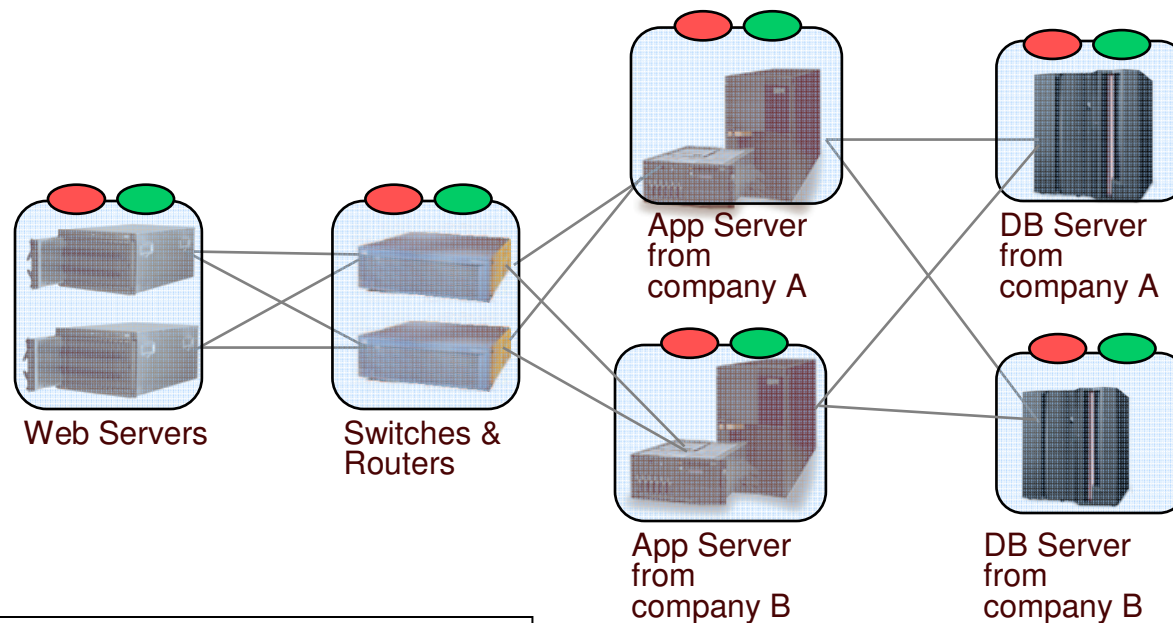
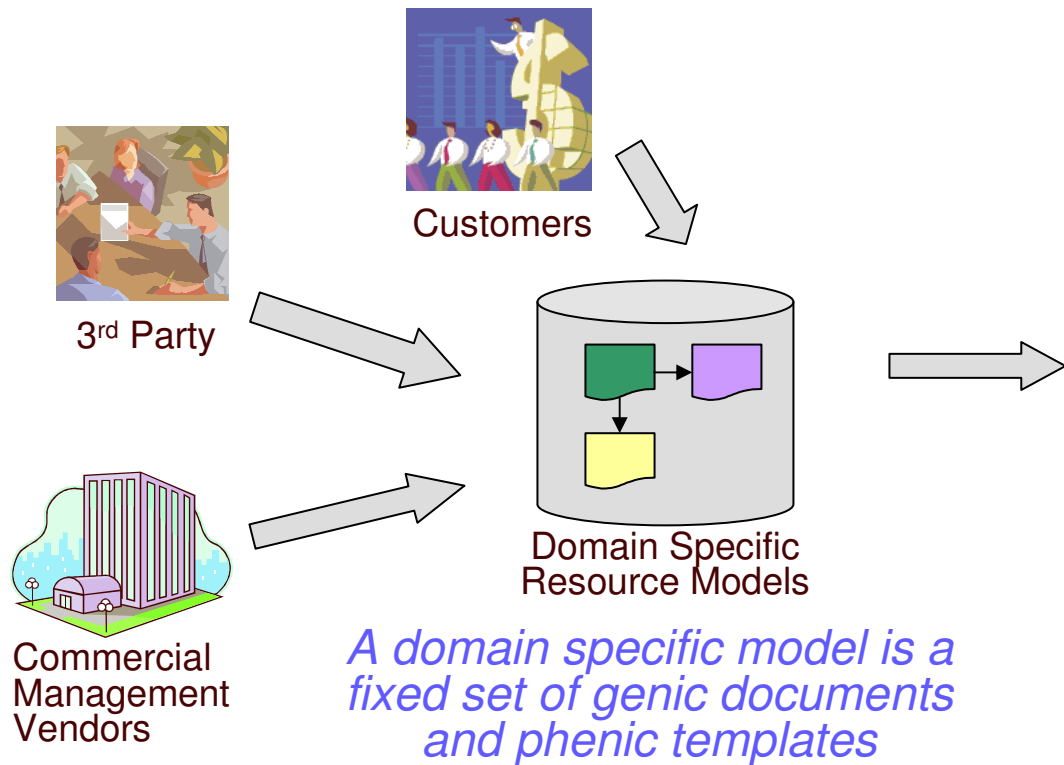The developer has no consistent way of looking at all the resources in their environment

No consistent approach to describing the important management information
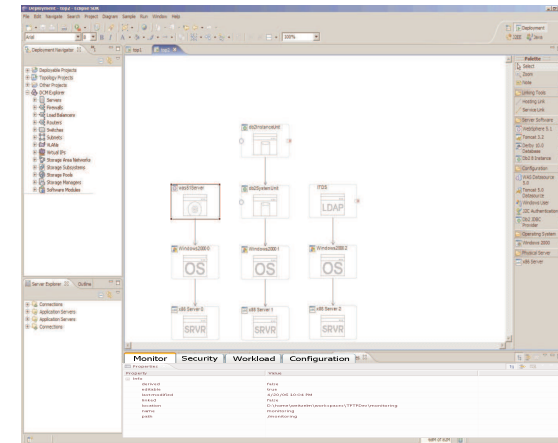
Developer

Customer Pain Points

- Difficult to extend existing instrumentation tooling for new roles or management disciplines

- New "kinds" of resources difficult to add

App Server from company A

DB Server from company A

Web Servers

Switches & Routers

App Server from company B

DB Server from company B

Vendor Specific Management APIs

Resource Modeling Landscape

Management Disciplines

Customers

3rd Party

Commercial Management Vendors

Domain Specific Resource Models

A domain specific model is a fixed set of genic documents and phenic templates

- Deployment
- Monitoring
- Security
- Problem Determination
- Performance
- Availability

Each discipline specific tool deals with additional genic constraints and phenic instances

From the SML Spec…

Genic Documents

The subset of documents in a model that describes the schemas and rules that govern the structure and content of the model's documents. This specification defines two kinds of genic documents - XML Schema documents that conform to SML's profile of XML Schema and rule documents that conform to SML's profile of Schematron.

Phenic Documents

The subset of documents in a model that describe the structure and content of the modeled entities.

Resource Model Value

- **Value to Resource Model providers**
  - Decouples the management tools from the resource providers

- **Value to domain model consumers**
  - Decouples from resource provider

- **Value to end user**
  - Vendor neutral, extensible, tooling based upon resource models
  - Deployment and Configuration of resources
  - Simple tools to create complex models
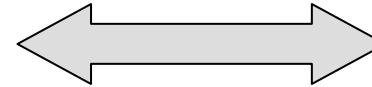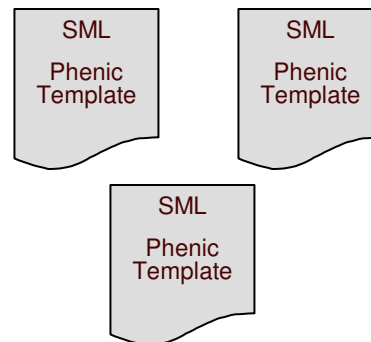 - Integration of Root Cause & Problem Determination

Requirements for Resource Model Ecosystem

- **Value increases exponentially with # resource models**

  - Enable creation of domain specific models

  - Demonstrate generic consumption as a building block

- **To make the models interoperable, they must be extensible**

  - Must be easy to extend

  - Must provide exemplary framework to enable rapid creation of role specific tooling
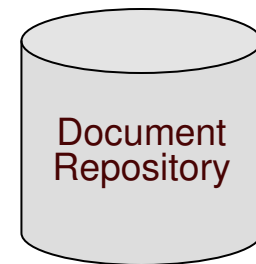
COSMOS deliverables related to the creation of domain specific models

- Tooling for validating SML instances and SML-IF documents

- Tooling for importing/exporting SML-IF documents to and from predefined repositories.

    - Import and export capabilities should be extensible so that consuming products can produce implementations to deal with custom repositories

    - Open source Document Repository is implemented as a file system structure

- Tooling for creating SML template documents

    - an SML template document is an SML instance defining a common pattern that can be re-used and adapted in different domain models

- Tooling for creating domain models based on existing SML templates (Resource model builder)

    - The tool should be extendable to allow registration of third parties SML templates

- Tooling required to create valid SML-IF documents. These may include both genic and phenic content

- "Templates" can be provided as phenic documents

- Extensible architecture to allow registration of new domain models or extensions to existing models

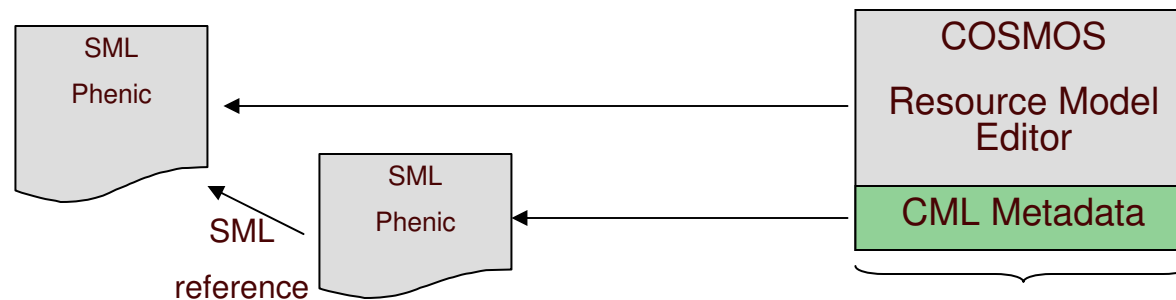SML Phenic Template

SML Phenic Template

SML Phenic Template

- Standard API for "export", "importing"

- Explore CMDB-f team's work on APIs to see if this is applicable

Document Repository

- Open source implementation is a file system

Using SML in the Monitoring life cycle:
Annotating what can be observed on a resource

- Resource Model Builder

  - Based on a set of SML template document (CML metadata)

  - Extendable to use third parties template documents

  - SML template documents are used to build SML phenic instances; the result is an SML-IF document

```
SML                              COSMOS
Phenic          ◄──────────    Resource Model
                                   Editor
          SML
        Phenic       ◄──────     CML Metadata
SML
reference
```
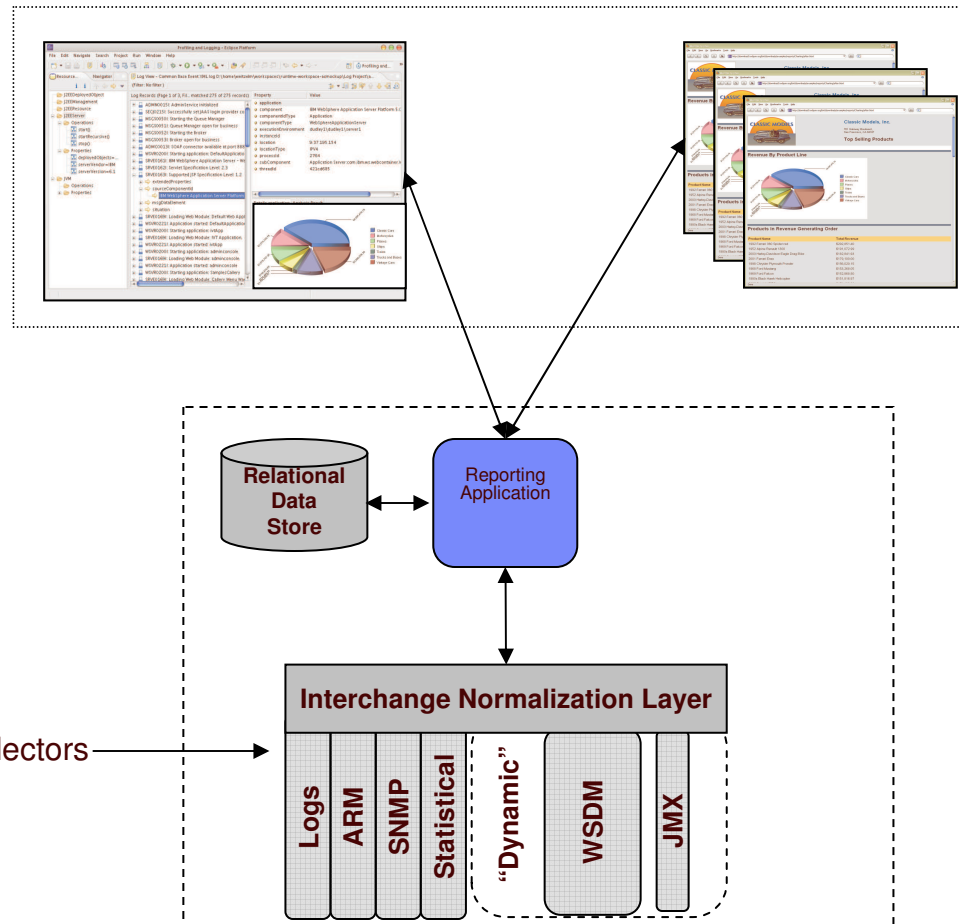
- This creates a SML phenic document that captures the monitoring metadata

- This is a CML proposal**

COSMOS: Key Components

- **Resource Modeling**

- **Build to Manage (BtM) a.k.a Enabling Management**

- **Monitoring**
  - Data collection
  - Reporting

Using BtM in the Monitoring life cycle:

Commercial monitoring systems can:

- Reuse COSMOS data collectors

Relational Data Store ↔ Reporting Application

Interchange Normalization Layer

Logs | ARM | SNMP | Statistical | "Dynamic" | WSDM | JMX

COSMOS: Key Components

- **Resource Modeling**

- **Build to Manage (BtM) a.k.a Enabling Management**

- **Monitoring**

  - Data collection

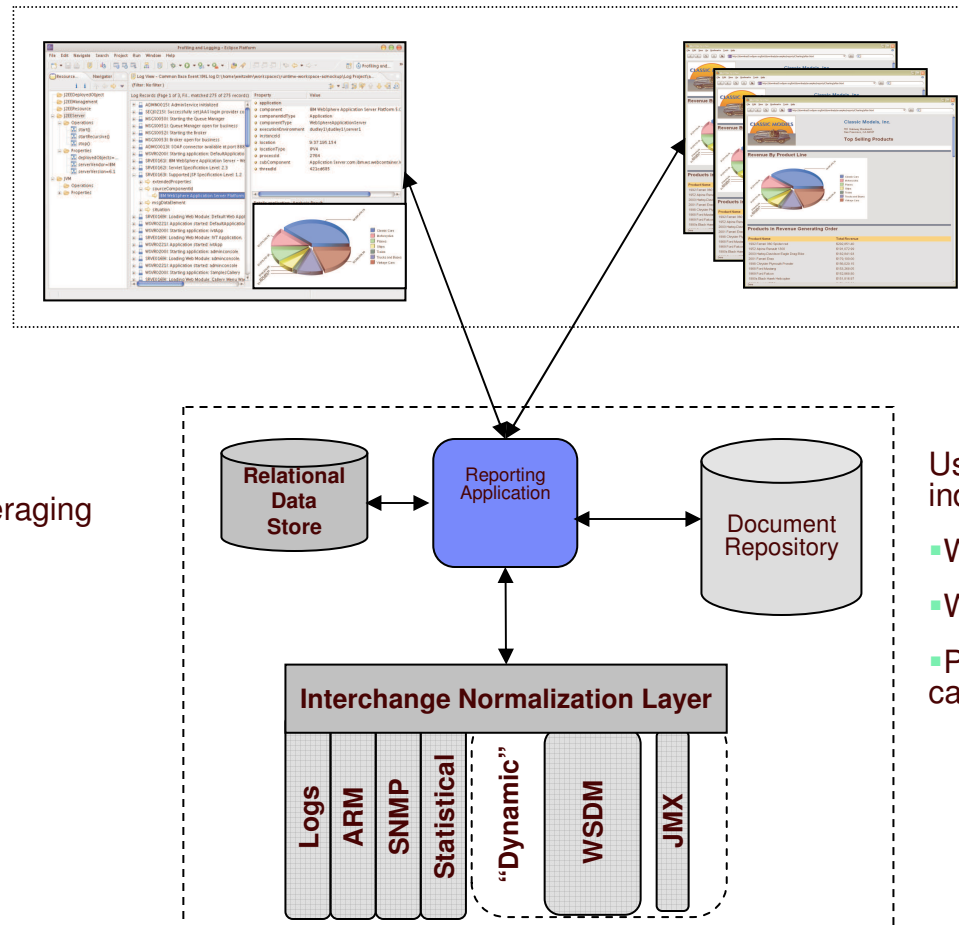 - Reporting

Using Data Collection & Reporting in the Monitoring life cycle:

▪Define a common set of reporting APIs

▪Use BIRT's web service/xml data source to encapsulate the database schema

Commercial monitoring systems can:

▪reuse community created reports

▪ extend their data by leveraging the content of the relational data store

Relational Data Store

Reporting Application

▪Define an API for the interchange normalization layer that the reporting application can use

Interchange Normalization Layer

Logs | ARM | SNMP | Statistical | "Dynamic" | WSDM | JMX

Using SML in the Monitoring life cycle:
Making the observation

Commercial monitoring systems can:

▪ extend their data by leveraging the document repository
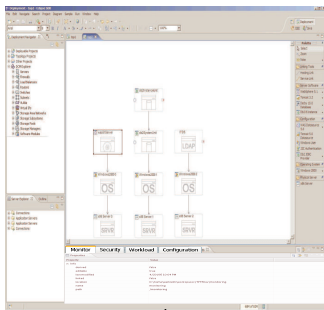
Use phenic documents to indicate

▪What to collect

▪What information is observable

▪Phenic documents used to carry the data
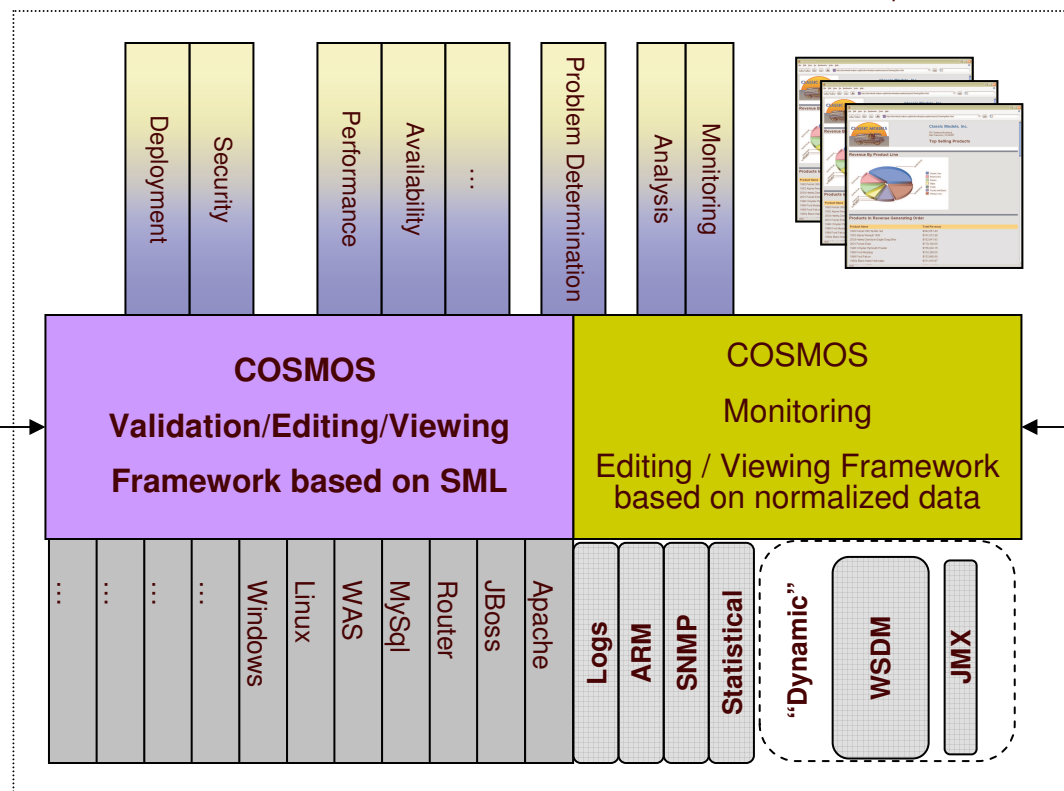
COSMOS Resource monitoring and modeling tools

COSMOS is committed to providing
exemplar usage of its APIs without eroding
commercial opportunities

Domain specific value add plug ins can be
completely in commercial space, with
minimal open source capability to
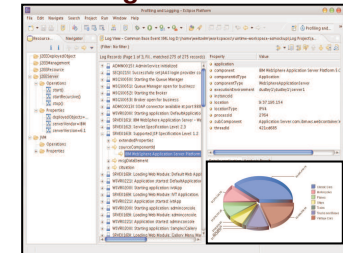demonstrate and prove the framework

**Resource Model
Builder**

**Build-to-Manage
Tooling**

Deployment | Security | Performance | Availability | ... | Problem Determination | Analysis | Monitoring

COSMOS

Validation/Editing/Viewing

Framework based on SML

COSMOS

Monitoring

Editing / Viewing Framework
based on normalized data

Common framework for
defining and expressing
the key instrumentation
points of the resource

... | ... | ... | ... | Windows | Linux | WAS | MySql | Router | JBoss | Apache | Logs | ARM | SNMP | Statistical | "Dynamic" | WSDM | JMX

Commercial
value add

:

Open
Source

Domain specific data collectors and resource models

Glossary

- **SML validation**
  - The action of validating SML extensions which are provided as additions to the xml 1.0
  - Schematron validation
- **SML-IF instance validation**
  - The process of validation the content of an SML-IF resource
  - Validate SML phenic documents contained by the SML-IF resource
  - Apply any schematron rule defined as a genic document
- **Template document**
  - An SML instance defining a common pattern that can be re-used and adapted in different domain models
- **Domain models**
  - The root of an SML-IF document. Contains a set of phenic and genic documents
- **Resource domain**
  - A set of genic and/or template documents that can be used to build define a domain
- **Template editor**
  - An editor that can create genic and template documents
- **Model editor**
  - An editor that can create domain model instances, based on a set of predefined templates

COSMOS Project Milestones – 2006-2007

2006	2007
September: ▪Kickoff in Eclipse and gain agreement on project charter ▪Establish infrastructure within Eclipse **October/November:** ▪Begin work on ▪SML based tooling ▪Data collection & visualization ▪Build to Manage **December:** ▪Begin incubation under guidance of Technology PMC	**January:** ▪Review SML tooling prototype at SML interoperability conference with partners. **March:** ▪Graduate/Launch COSMOS as Top level project ▪COSMOS Release 1 Availability ▪SML based tooling ▪Data collection & visualization **June: Europa** ▪COSMOS release 1.5 Availability ▪Monitor UI ▪Remainder of Data Collection ▪Additional SML tooling updates **November & December:** ▪Tbd

Comments about the community response to the proposal

- General agreement across community members that current proposal is "spot on".   Lets move forward!

  - Next step should focus on high level use cases and requirements

- Lets make sure we understand what other Eclipse projects have technology we can leverage and where we need to provide new code

- Lets make sure we can link back to the standards being supported and any implementations if applicable

  - Eg. Resource modeling will be in support of SML

Initial Community

The following companies have actively participated in shaping the project and may contribute committers once the project is formed:

- **Alterpoint (www.alterpoint.com)**

- **Cisco (www.cisco.com)**

- **Compuware (www.compuware.com)**

- **GroundWork (www.groundworkopensource.com)**

- **Intel (www.intel.com)**

- **IBM (www.ibm.com)**

- **OC Systems (www.ocsystems.com)**

- **Simula Labs (www.simulalabs.com)**

- **Sybase (www.sybase.com)**

Confirmation that the project members have read and understand the Eclipse Development process and these guidelines

- http://wiki.eclipse.org/index.php/Development_Process_2006_Revision

- **This is the new one that Bjorn is working on**

Glossary

- **SML validation**
  - The action of validating SML extensions which are provided as additions to the xml 1.0
  - Schematron validation
- **SML-IF instance validation**
  - The process of validation the content of an SML-IF resource
  - Validate SML phenic documents contained by the SML-IF resource
  - Apply any schematron rule defined as a genic document
- **Template document**
  - An SML instance defining a common pattern that can be re-used and adapted in different domain models
- **Domain models**
  - The root of an SML-IF document. Contains a set of phenic and genic documents
- **Resource domain**
  - A set of genic and/or template documents that can be used to build define a domain
- **Template editor**
  - An editor that can create genic and template documents
- **Model editor**
  - An editor that can create domain model instances, based on a set of predefined templates