The Ohio State University Medical Center

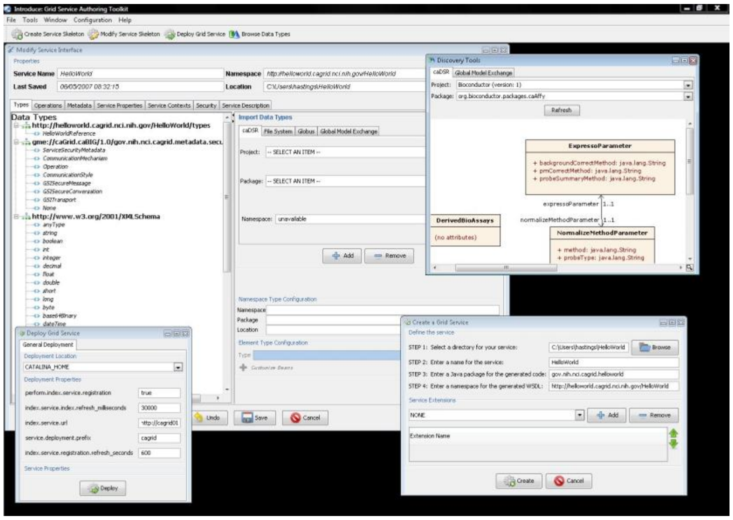
Introduce

http://www.cagrid.org

Grid Service Authoring Toolkit

Introduce aims to reduce the service development and deployment effort by hiding low level details of the Globus Toolkit and to enable the implementation of strongly-typed Grid services. We expect that enabling strongly typed grid services while lowering the difficulty of entry to the Grid via toolkits like Introduce will have a major impact to the success of the Grid and its wider adoption as a viable technology of choice, not only in the commercial sector, but also in other areas such as academic, medical, and government research.

Service-oriented architectures, standards, and applications have gained wide acceptance in the Grid computing community. A number of tools and middleware systems have been developed to support application development using Grid Services frameworks. Most of these efforts, however, have focused on the low-level support for management and execution of Grid services, management of Grid-enabled resources, and deployment and execution of applications that make use of Grid services. Simple-to-use service development tools, which will allow a Grid service developer to leverage Grid technologies without needing to know low-level details, are becoming increasingly important. Moreover, support for development of strongly-typed services, in which data types consumed and produced by a service are well-defined and published in the Grid, is necessary to enable syntactic interoperability so that two Grid endpoints can interact with each other programmatically and correctly.



Properties

Service Name: HelloWorld

Namespace: http://helloworld.cagrid.nci.nih.gov/HelloWorld

Last Saved: 06/05/2007 08:32:15

Location: C:\Users\hastings\HelloWorld

Data Types

- http://helloworld.cagrid.nci.nih.gov/HelloWorld/types
  - HelloWorldReference
- gme://caGrid.caBIG/1.0/gov.nih.nci.cagrid.metadata.secu
  - ServiceSecurityMetadata
  - CommunicationMechanism
  - Operation
  - CommunicationStyle
  - GSISecureMessage
  - GSISecureConversation
  - GSITransport
  - None
- http://www.w3.org/2001/XMLSchema
  - anyType
  - string
  - boolean
  - int
  - integer
  - decimal
  - float
  - double
  - short
  - long
  - byte
  - base64Binary
  - dateTime

Import Data Types (caDSR | File System | Globus | Global Model Exchange)

Project: -- SELECT AN ITEM --

Package: -- SELECT AN ITEM --

Namespace: unavailable

Namespace Type Configuration: Namespace, Package, Location

Element Type Configuration: Type; Customize Beans

Discovery Tools (caDSR)

Project: Bioconductor (version: 1)

Package: org.bioconductor.packages.caAffy

ExpressoParameter
+ backgroundCorrectMethod: java.lang.String
+ pmCorrectMethod: java.lang.String
+ probeSummaryMethod: java.lang.String

expressoParameter 1..1

normalizeMethodParameter 1..1

NormalizeMethodParameter
+ method: java.lang.String
+ probeType: java.lang.String

DerivedBioAssays
(no attributes)

Deploy Grid Service

Deployment Location: CATALINA_HOME

Deployment Properties

perform.index.service.registration	true
index.service.index.refresh_milliseconds	30000
index.service.url	ttp://cagrid01
service.deployment.prefix	cagrid
index.service.registration.refresh_seconds	600

Service Properties

Create a Grid Service

Define the service

STEP 1: Select a directory for your service: C:\Users\hastings\HelloWorld

STEP 2: Enter a name for the service: HelloWorld

STEP 3: Enter a Java package for the generated code: gov.nih.nci.cagrid.helloworld

STEP 4: Enter a namespace for the generated WSDL: http://helloworld.cagrid.nci.nih.gov/HelloWorld

Service Extensions: NONE

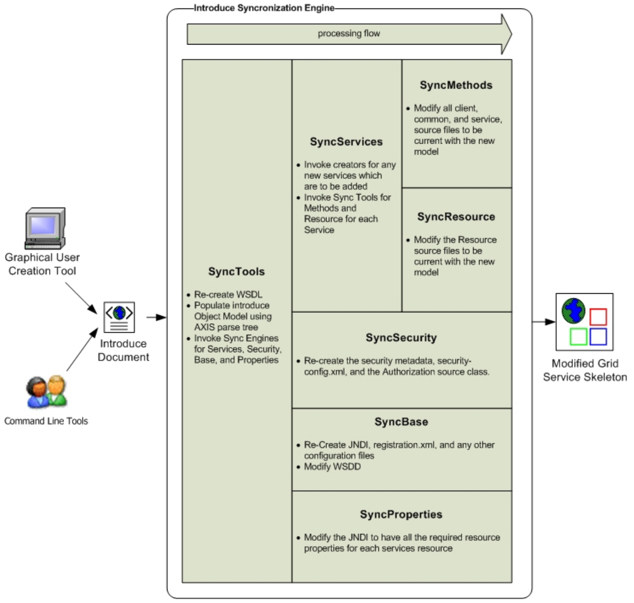
Extension Name

Features

- Graphical creation of all aspects of the grid service from describing operations and stateful resources to configuring complex authentication and authorization policy.
- Dynamic schema discovery enabling use of published data types as service parameters and resource properties.
- Creates WSDL2.0 / WSRF Compliant Services utilizing Globus 4.0.X and Axis for grid service middleware support.
- All client and service code is automatically generated using the Introduce Synchronization engine as well as Axis and JET.
- Supports Creating Multiple Resources/Services using the WSRF family of specifications.
- Utilizes Globus GSI Security and provides graphical Configuration of authentication and authorization policies.
- CSM and GridGrouper Authorization Support.
- Resource Properties can be used for service metadata and published using MDS based Index Service Registration.
- Rich extension/plug-in framework for creating custom services or adding custom functionality to Introduce.
- Supports WS-Notification, WS-Lifetime, Persistent Resources, Secure Resources as well as many others.

Architecture

- Service Model maintains the complete representation of the services, methods, resources, resource properties, data types, security configuration. Index service registration, etc.
- Graphical Development Environment modifies service model and passes the modified model to the Introduce Synchronization Engine.
- Introduce Synchronization Engine re-syncs/generates the service with changes in the modified service model.
- Synchronization engine leverages Java Emitter Templates (JET), Axis 1.2, and Globus 4.0.X for creation and modification of source code and configuration files required for the service
- Auto generation of unboxed complete client and stubbed service implementation.
- Globus Policy Decision Point (PDP) is automatically generated and hooked in based on the developers described security constraints.
- Full support for stateful services utilizing factory patterns for dynamic creation of WSRF service resources.


Introduce Syncronization Engine
processing flow
Graphical User Creation Tool
Command Line Tools
Introduce Document
Modified Grid Service Skeleton

SyncTools
- Re-create WSDL
- Populate introduce Object Model using AXIS parse tree
- Invoke Sync Engines for Services, Security, Base, and Properties

SyncServices
- Invoke creators for any new services which are to be added
- Invoke Sync Tools for Methods and Resource for each Service

SyncMethods
- Modify all client, common, and service, source files to be current with the new model

SyncResource
- Modify the Resource source files to be current with the new model

SyncSecurity
- Re-create the security metadata, security-config.xml, and the Authorization source class.

SyncBase
- Re-Create JNDI, registration.xml, and any other configuration files
- Modify WSDD

SyncProperties
- Modify the JNDI to have all the required resource properties for each services resource

SRI SOFTWARE RESEARCH INSTITUTE

questions or comments
hastings@bmi.osu.edu