Agenda Today (Tuesday)

- 8:00 – 9:00 Overview, Next Version
- 9:00 – 9:30 Methods
- 9:30 – 10:30 – Overview, SLP implementation
- 10:30-10:45 – Break
- 10:45 -12:30 – Indications
- 12:30- 13:30 – Lunch
- 13:30 – 14:30 – CMPI overview
- 14:30 – 15:45 Associations Overview
- 15:45 – 16:00 Break
- 16:00 – 17:30 Security
- 19:00 – Infinity (22:00) - Lab
Agenda Wednesday

- 9:00-19:30 – Pegasus processes and procedures
- 10:45-12:15 – Open Pegasus Architecture Team Meeting
- 13:15-14:45 – Pegasus Platform Maintainers Discussion
- 15:00 – 16:30 Steering Committee meeting
- 16:30 – 17:00 Conference Wrap-up
Agenda for 2.3 Overview

- Overview of Versions in Pegasus
- Schedule and Plans for Next Version
- Overview of major new functionality in next version
Pegasus Versions

- New Functionality set by Pegasus Project
  - Driven by needs of users (Those who communicate there needs) and who can do the work.
- Typically new point releases about every 6 months.
- There is a defined process (to be discussed Wednesday morning)
Major Functionality By Version

Version 1.0 – August 2001
- Single Threaded
- Included basic components (cimom, repository, client, providers).
- Initial version of almost all C++ objects
- Executed major operations but no indications, number of errors in operation execution, etc.

Version 2.0 – July 2002
- Completed multitasking work
- Completed most operations
- Indications basic code installed
- Some operations not complete
- APIs not frozen

Version 2.1 – Nov 2002
- Provider/Client APIs frozen
- Problem cleanup

Version 2.2 – June 2003
- Completed All operations (except Query)
- Indications in place in Server
- Modified Routing Functionality
- Installed Formal Procedures for control
- Reshaped Provider Manager
- Production oriented Logging
- CIM 2.7
- Misc Cleanup
  - CIMOM Handle
  - Communication Processing
General Milestones for a Pegasus Release

- Start
- RSO – Requirements Sign-off
- Functionally Complete (FC)
- Certification Test Complete (CTC)
- Ship Release (SR)
Major Goals for Version 2.3

- Out-of-Process Providers
- Pluggable Provider Managers
- CIM Listener Client API
- Operation Performance Enhancement
- Security Plugin Enhancements
- Implement much of Interop Classes
- SLP Discovery *
- Initial Globalization *
- CMPI Provider Manager * (CMPI Interfaces)
- Freeze Association Provider Interfaces

* Subjects that will be covered later today
Version 2.3, Dates

- Started – May 2003
- RSO – 6/30 planned 7/15 done
- Functionally Complete (FC) – 8/15
- Certification Test Complete CTC 10/10/03
- Ship Release – 10/17/02

- See PEP 57 for more information
Out-of-Process Providers

- Eventual Goal is
  - Out of process and out-of-system providers
- Version 2.3
  - First cut at out-of-process providers
- Functionality
  - Initial implementation – Probably all providers out of process.
  - Later – Mixed in and out of process.
  - This is in discussion in Architecture Team now.
Pluggable Provider Managers

- **Requirement**
  - Support multiple languages for provider writing
  - Support multiple models for provider control and loading
  - Allow provider controllers to plug into Pegasus without reworking core

- **Short Term Requirement**
  - Provide CMPI C interfaces for Pegasus

- **Longer term view**
  - Java, Perl, etc. as Provider languages
Provider Managers

- Provider Managers map the base provider interface onto one or more other provider interfaces. This scheme may be used to:
  - Refine the base provider interface into several provider interfaces (as mentioned above).
  - Convert from message based operations to direct call for the provider.
  - Map the provider interface onto other language provider interfaces (e.g., Java, Perl, Tcl).
Provider Managers

- **Objectives**
  - Pluggable
  - Clear interface to CIMOM
  - Manage individual providers
  - Manage provider registration

- **Uses**
  - Provider Language mapping
  - Remote provider implementation
  - Extend providers to other pluggable functionality
  - API versioning.

- **Planned Languages**
  - C (CMPI)
  - Java
CIM Listener API

- Requirement
  - Define C++ API for receiving Indications in CIM Listener

- Status
  - Infrastructure components existed (HTTP, decoder, etc.)
  - In process now
Operation Performance Enhancements

- Define some limited performance enhancements for operations
  - Ex. Caching, optimization of association lookup, etc.
Implement Interop Class

- Requirement
  - Implement Interop Classes to provide info for
    - Capabilities requests
    - SLP information

- Implement functions of Interop Class required for SLP
  - CIM_Namespace
  - CIM_ObjectManager
  - CIM_ObjectmanagerCommunicationMech
  - CIM_ProtocolAdapter
  - Associations

- This does not include CIM_Provider group.
SLP Discovery

- Implement in accord with the planned specification
  - Client API
  - User Agent for Client
  - Service Agent
  - Information access for SLP template from CIM Information (mostly Interop Classes).

- More information in next presentation
Internationalization

- Requirement
  - Implement operation internationalization
    - CIMOM itself
    - Internationalization info for providers
    - Message catalogs for internationalized messages

- Further information this afternoon
CMPI Provider Manager

- Implement provider manager that will load an execute C language providers based on the CMPI definitions being defined by IBM
- CMPI specification to be published by OpenGroup
  - Work of IBM Germany, WEBMSource group, and Open Group
- Pegasus to implement this interface
Freeze Association Provider Interfaces

- Pegasus controls interfaces and has frozen most publicly available interfaces
  - CIMClient, Objects, Instance and method providers
  - Association is experimental today

- Goal
  - Complete any modifications to the provider interfaces and freeze them

- Status
  - There are a couple of possible minor “tweaks” that are being proposed to the Architecture team now.
After Version 2.3

- Early planning starting for 2.4
  - Wishlist and user requirements gathering
  - Will seriously start planning about time of Version 2.3 Functional Complete
- We want to work on wish list in discussion tomorrow.