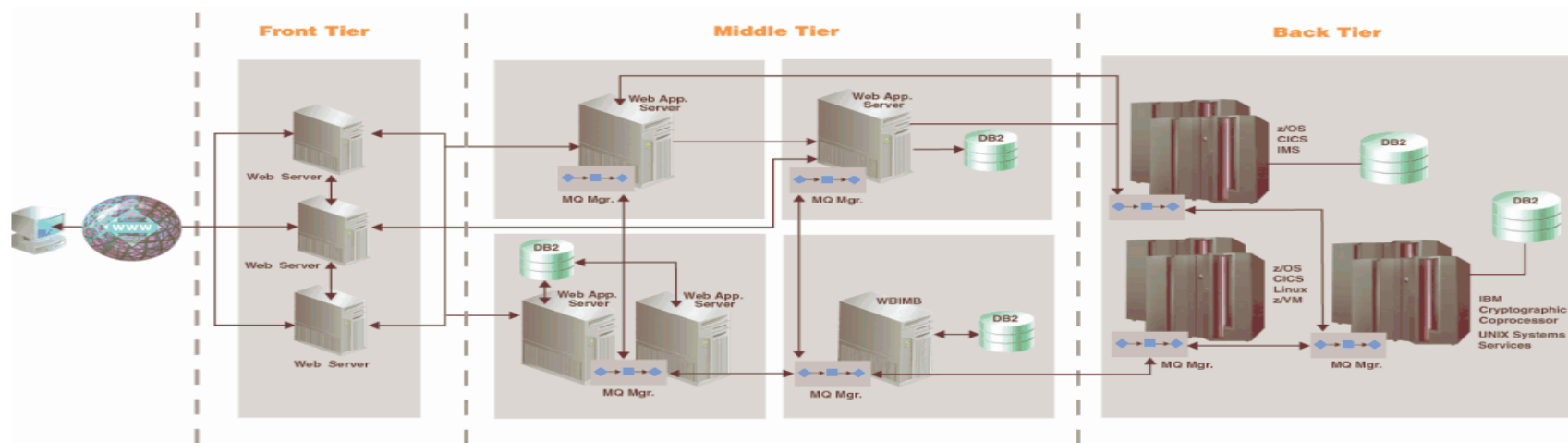


WSDM Build to Manage

Put together by Mark Weitzel
weitzelm@us.ibm.com

Challenges of IT Systems Management

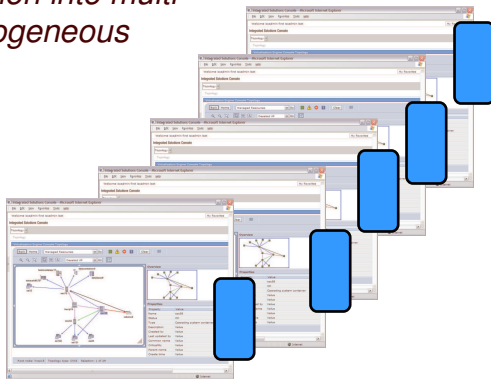
- Business processes increasingly depend on multi-tier composite applications
- Challenging to design, build, test, and manage for high performance and availability
- SOA, while introducing architectural flexibility exposes more inter-dependencies
- Traditional development and management processes need to be expanded to account for challenges



WSDM is for Simplification of Management Interfaces

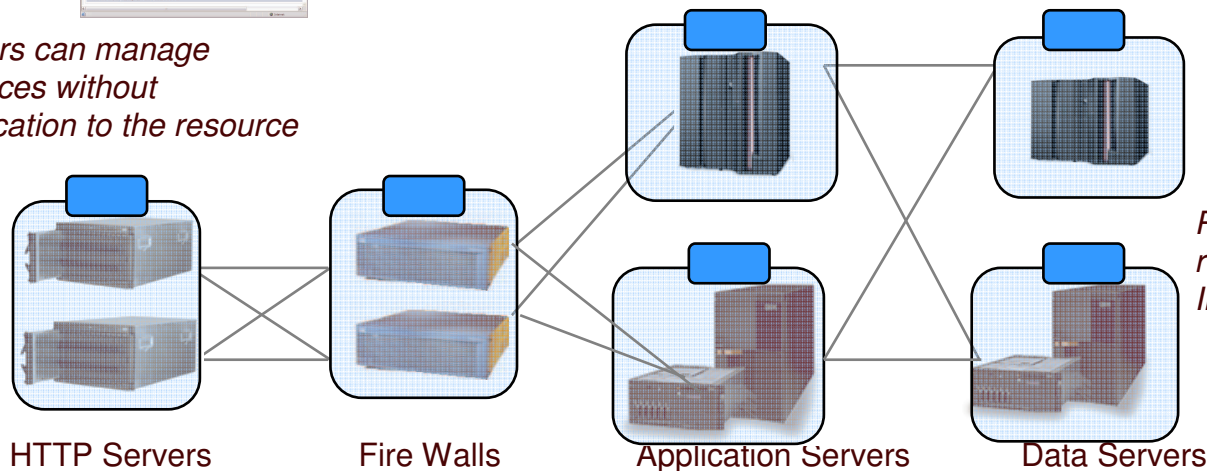
Value Proposition: WSDM provides the ability to expose management capabilities using Web services

Ease integration into multi-vendor heterogeneous environment



Eliminates the need to test compatibility or ship special adapters to work with vendor specific management applications

Vendors can manage resources without modification to the resource



- Leverage open standards to reduce infrastructure
- Change is easier with standard semantics for application behavior.
- End-to-end problem determination accelerated by common data format.

 **WSDM Standard Management API (*)**

Resources can be managed regardless of Internal Instrumentation

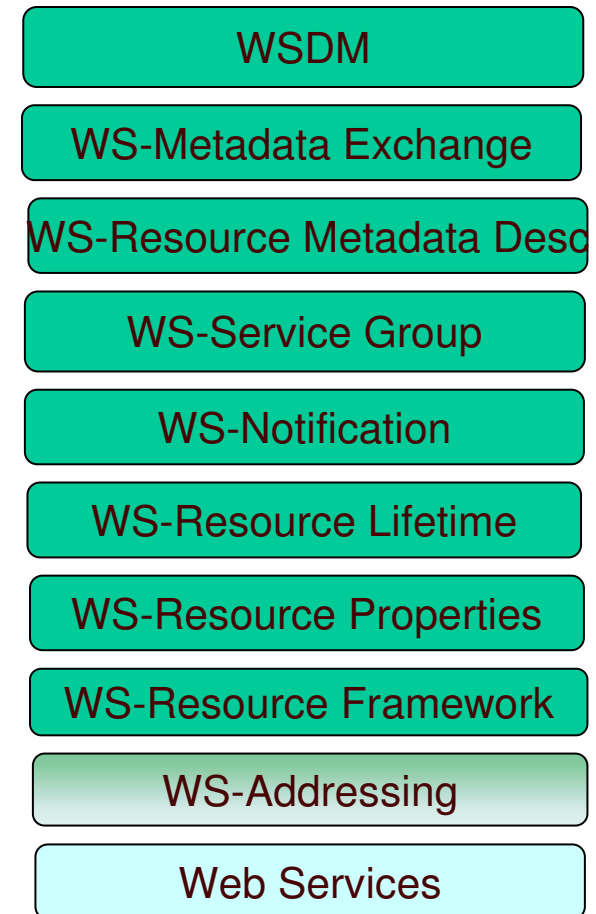
*WSDM is a standard defined by OASIS, and is currently at the 1.1 level.

What is WSDM?

- **Web Services Distributed Mangement**
 - Management USING Web Services (MUWS)
 - Web services to describe and access manageability of resources
 - Management applications use Web services just like other applications use Web services
 - Management OF Web Services (MOWS)
 - An application of Management Using Web Services for the Web Service as the IT resource
- **Leverage Web services foundation to enable interoperability between managers and manageable resources**
 - Interoperability *between* managers
- **Ratified standard from Oasis**
 - Current version is 1.1

WSDM Family of Specifications

- Built on top of the WS-Resource Framework
 - WS-Resources are *stateful* web services
 - WS-Addressing used to indicate the *specific instance* of a service
 - End Point References (EPR)
- WSDM introduces standard set of Manageability Capabilities
 - Identity, Manageability Characteristics, Operational Status, Metrics, ...
 - Provides a common set of semantics for all resources



Exploiter Value of WSDM

- **Deploy new resources quickly**
 - **Leverages web services techniques for service announce/discovery**
 - **Management system dynamically adjusts to business demands**
 - **Maintains critical management despite change in the underlying infrastructure**
 - **Reduce deployment and integration cost**
 - **Reduces Manual configuration activity associated with the management domain**
 - **Facilitates integration from multiple vendors**
-

Industry Benefit of WSDM

- ❖ Consistent approach for managing SOA Applications and the IT resources that support them
- ❖ Flexibility for placement of management function
 - close to the resource for efficiency
 - centralized for broader view of IT resources in a business context
- ❖ Facilitates interoperability between management products
- ❖ Enables management of an expanded and future set of standards-based resources
- ❖ Reduces TCO for operations management products making them more competitive
 - Reduces complexity of changing the management system when the underlying infrastructure changes

Apache Muse

- **History**

- IBM contributed its implementation of WSDM and made it available to the open-source Apache Muse Project
- We are supporting code portability that enables the WSDM implementations to run on different Web Services runtimes
 - Application Servers (Tomcat/WAS/etc..), OSGi

- **Reference Implementation of WS-Distributed Management 1.1 including**

- WS-Addressing 1.0, WSRF 1.2, WS-N 1.3, WS-RMD 1.0 and WS-MEX
- WSDM Event Format (WEF) 1.1 library

- **Implementation of Service Groups, Advertisements, Notifications, Subscriptions, resource creation, persistence mechanism**

- **Programming model that allows runtime composition of user-defined capability**

- **Runtime support for Axis-2, JAX-WS and Equinox/OSGi (on J2SE)**

- Working on OSGi Foundation Profile

- **Command line code generation utility for generating code, packaging, deployment descriptors and client proxies**

- **Ability to restrict customization of the runtime from end-user modification**

- **Utility functions/library for working with WSDM resources (dynamic proxy, resource inspector, simple analyzer)**

- **Roadmap for evolution to WS-UM**

- Initial implementation of WS-Resource Transfer

Build to Manage for WSDM: Overview

- **Eclipse tooling for generating WSDM manageability endpoints (MEPs)**
 - Often used to “wrapper” existing management interfaces, e.g. JSR77
- **Targeted towards the Developer**
 - Define the manageability interfaces
 - Implement & test the behavior
- **Leverages Apache Muse “under the covers”**
 - Generates java ‘stubs’ based upon Muse programming model
 - Independent of SOAP engine
- **Code deployable into multiple environments**
 - as WARs
 - as OSGi bundles
- **Browse endpoints with the Managed Agent Explorer**
- **Event notification support for property changed events**

Basic Scenario: Building a MEP

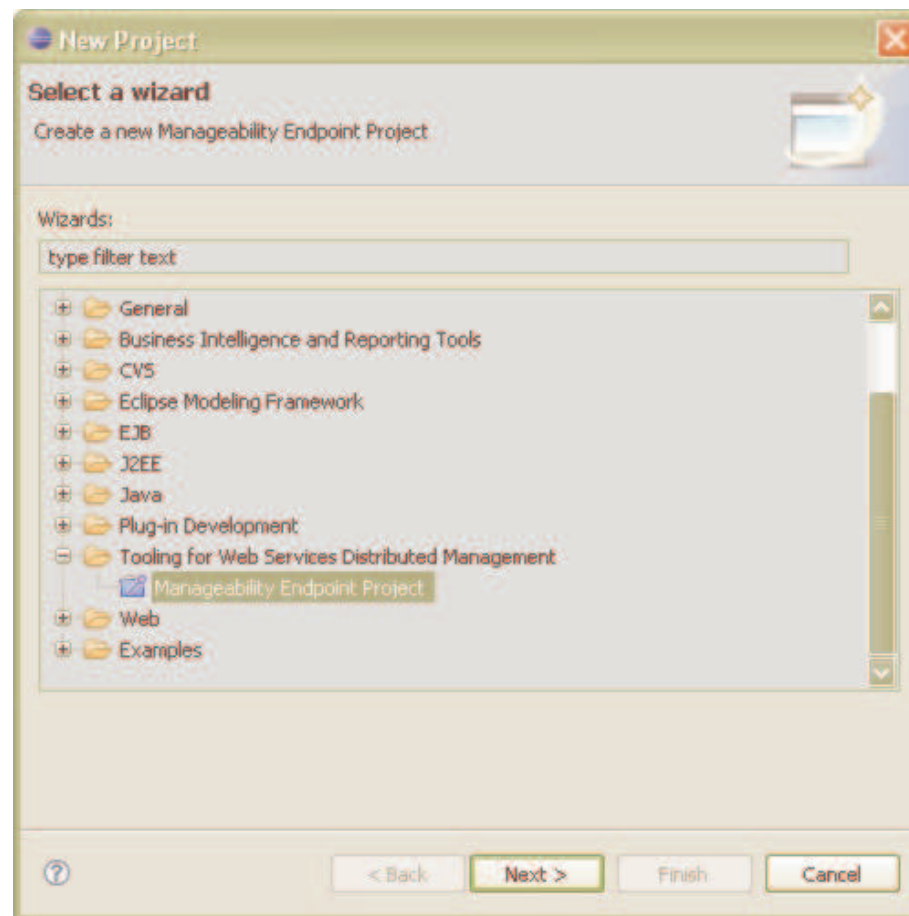


Developer

- **Create a Manageability Endpoint Project**
 - Holds the definition of the user defined capabilities and endpoint type
- **Create a new Manageability Endpoint Type**
 - Select the set of pre-defined capabilities
- **Create “custom” capabilities**
 - Properties / Operations / Events (topics)
 - Metadata
- **Generate the Java code**
 - Implement Manageability “Stubs”
- **Use MAX to help Test/Debug**

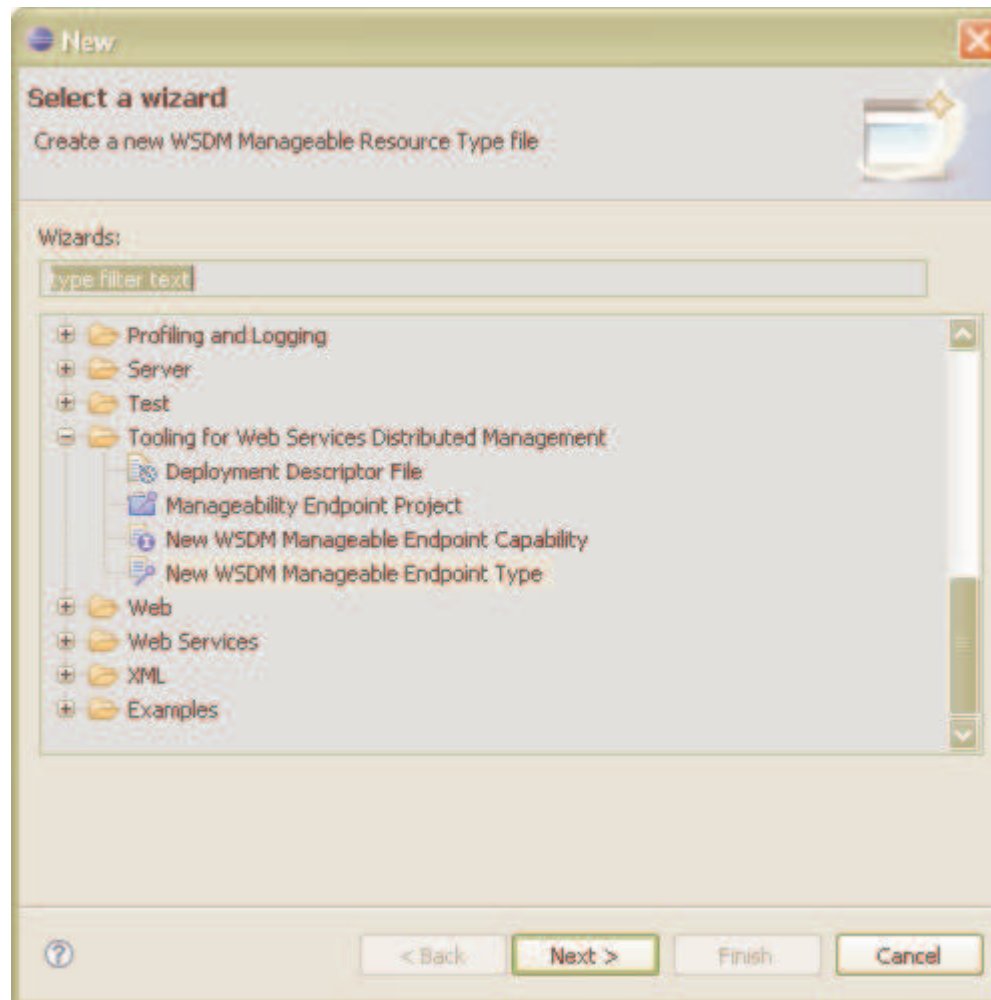
Basic Scenario

Create a Manageability Endpoint Project



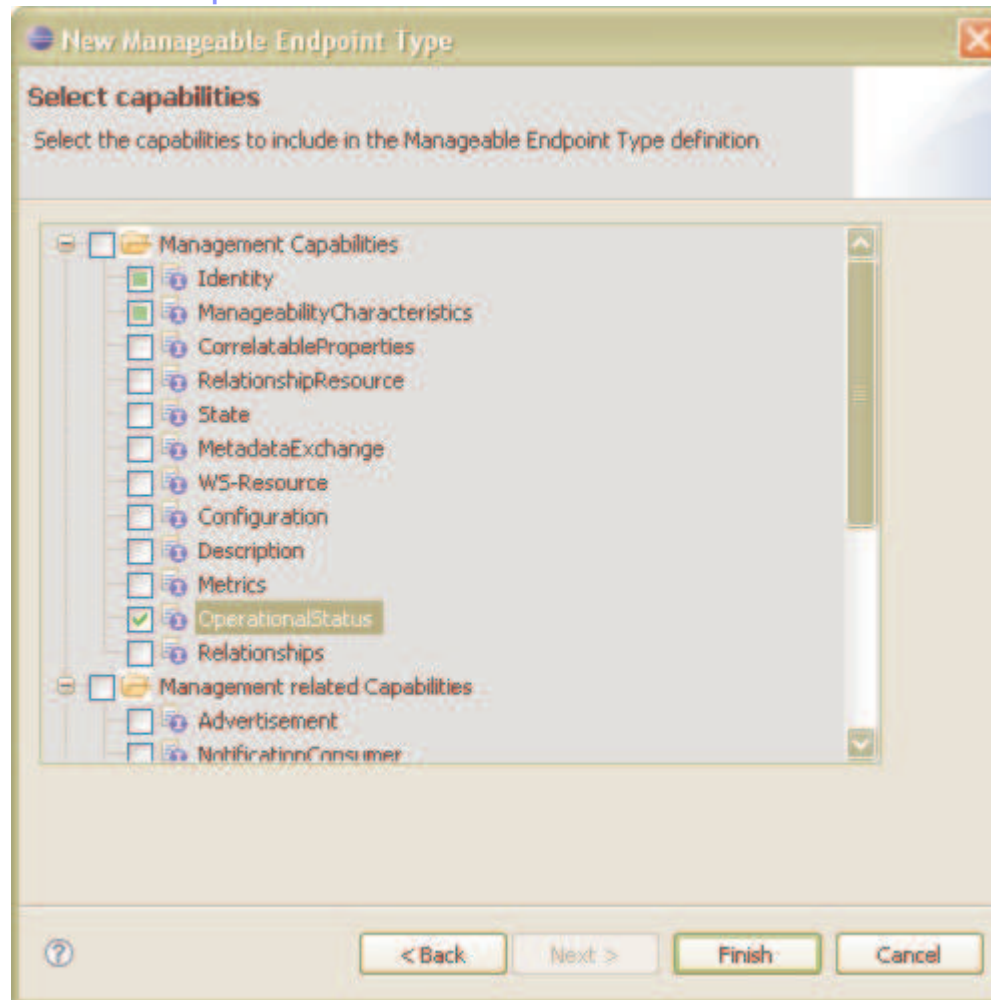
Basic Scenario

Create a new Manageability Endpoint Type



Basic Scenario

Select the set of pre-defined capabilities



Basic Scenario

Create “custom” capabilities

The screenshot shows the 'Property' dialog box for a file named 'ShipperCap.ncap'. The dialog is divided into two main panes: 'Properties' on the left and 'Details' on the right. The 'Properties' pane shows a list of 'Following properties used:' with one entry, 'status'. Below this list are three buttons: 'Add...', 'Edit metadata', and 'Remove'. The 'Details' pane contains several sections: 'Details of the selected property:' with fields for 'Property name:' (value: 'status') and 'Property type:' (value: 'string'); 'Make this property:' with radio buttons for 'Writable' (selected) and 'Read-only'; 'Mutability' with radio buttons for 'Mutable' (selected), 'Constant', and 'Appendable'; 'Cardinality' with radio buttons for 'Always present' (selected), 'Optional', '[0..unbounded]', '[1..unbounded]', and 'Other' (with input fields for '5' and '10'); and 'Metrics' with a table for 'Metrics of the selected property:'. The table has columns for 'Change Type', 'Time Scope', 'Gathering Time', 'Calculation Int...', and 'Metric Group'. At the bottom of the dialog is a navigation bar with tabs for 'Overview', 'Properties', 'Operations', and 'Topics'.

- **Resource Properties are added along with their metadata**

Basic Scenario

Create “custom” capabilities

*ShipperCap.mcap

Operation

Operations

Following operations used:

- suspend

Add... Remove

Details

Details of the selected operation:

Operation name: suspend

Return type: string

Import complex data types

Parameters

The following parameters are defined for the selected operation:

Name	Type
------	------

Add... Edit... Remove

Exceptions

The following exceptions are defined for the selected operation:

Type
BaseFaultType

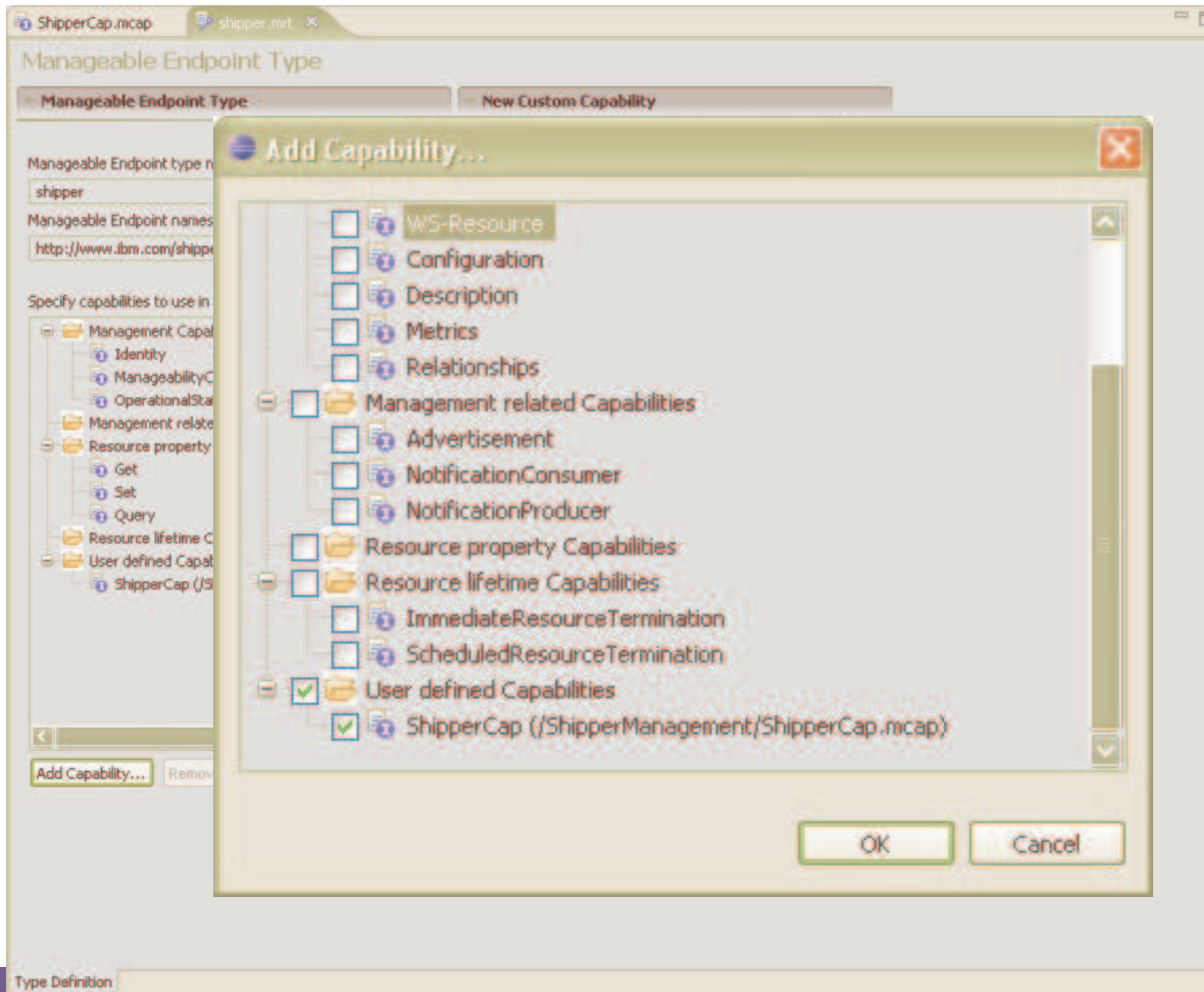
Add... Remove

Overview Properties Operations Topics

- **Management Operations** are declared, along with parameters and faults

Basic Scenario

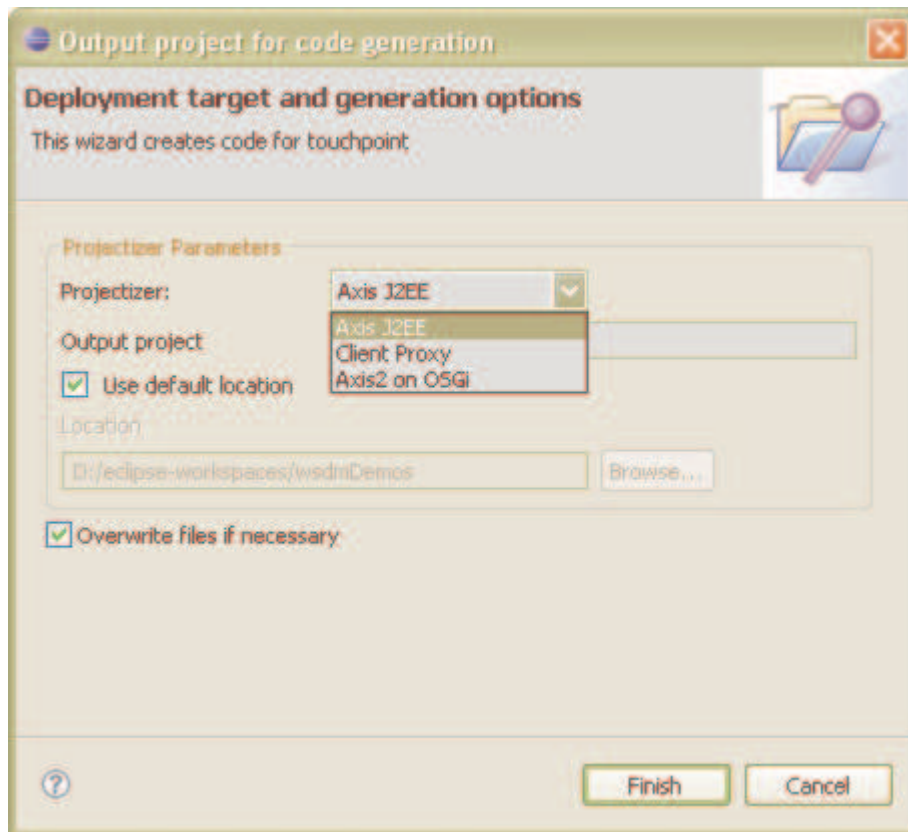
Create “custom” capabilities



Custom capability added to the manageability endpoint type

Basic Scenario

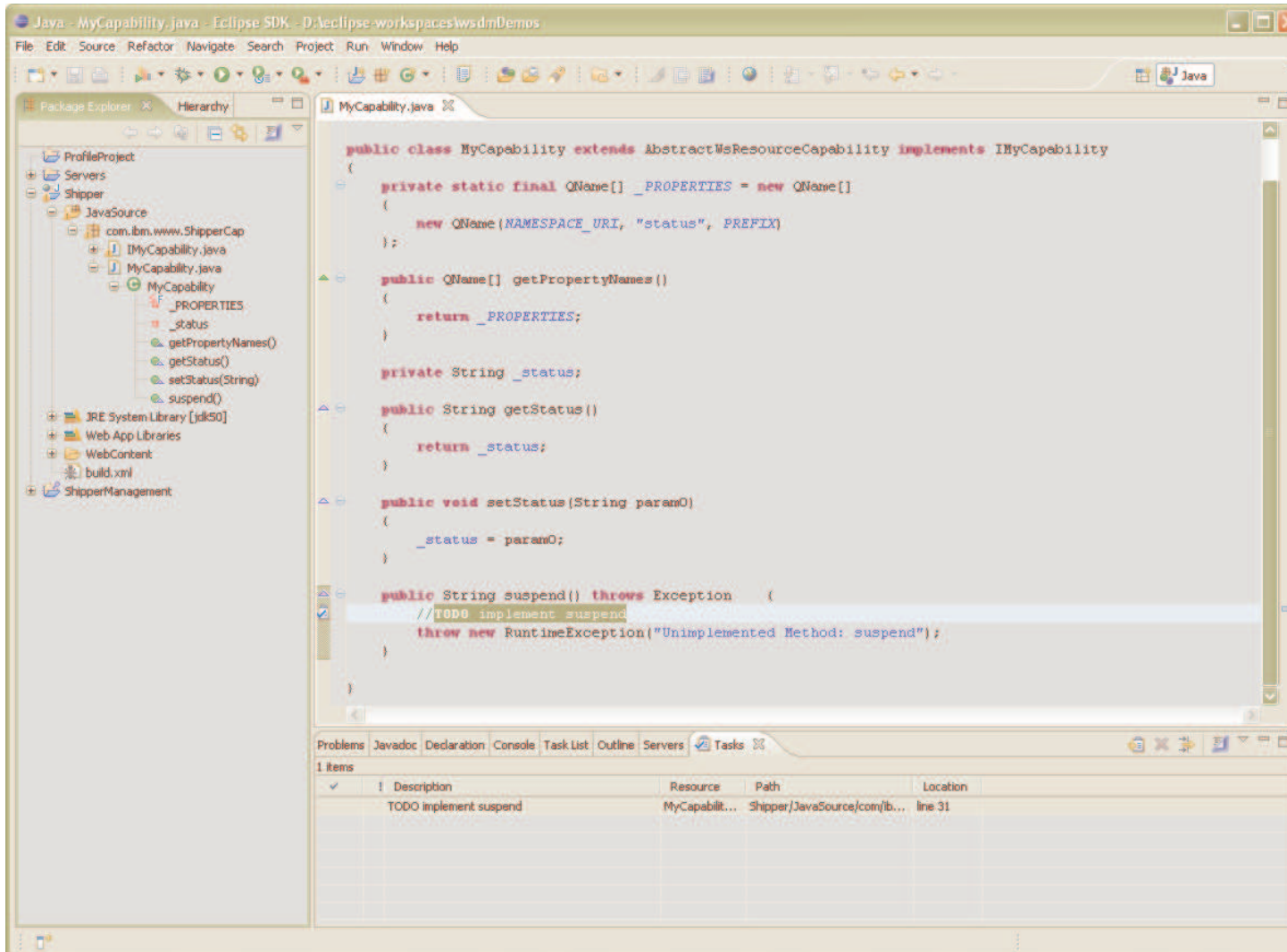
Generate the Java code



- **Multiple generation options include OSGi**

Basic Scenario

Implement the Java code



Each operation implemented with “TODO Task Marker”

Each property given “bean like getters/setters”

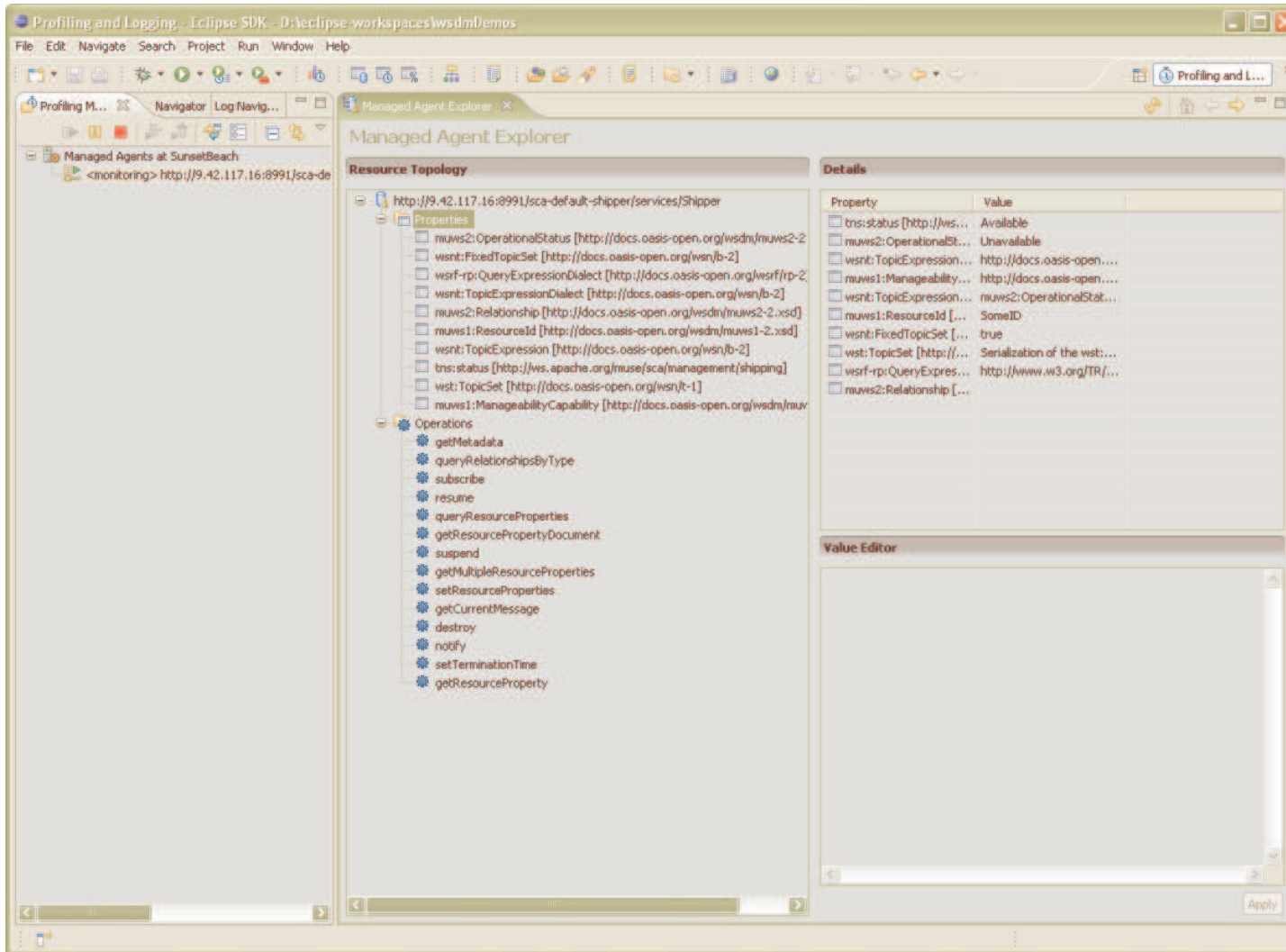
Web App created automatically

Integrated with “Run on Server”

- Automatically brings up MAX

Basic Scenario

Test using MAX



Integrated with TPTP profile view

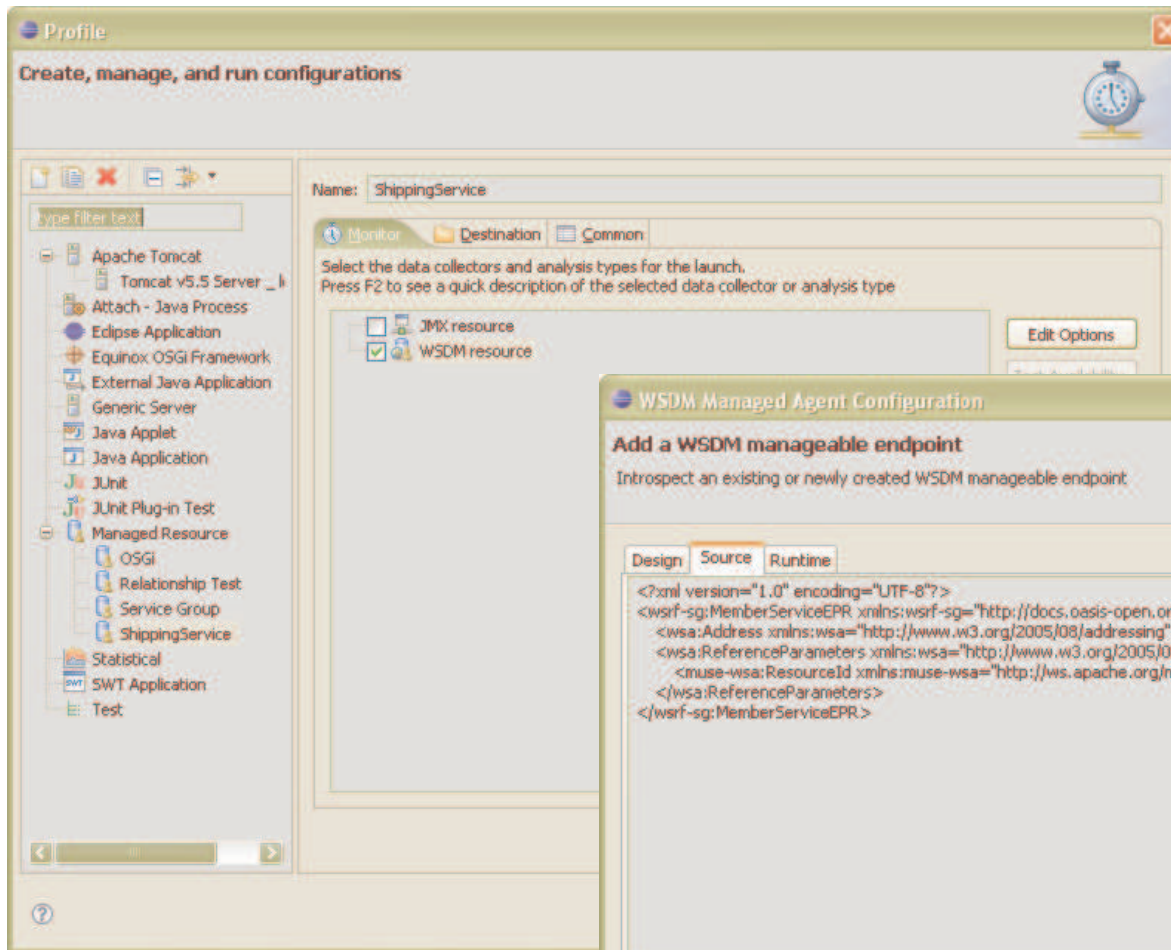
View/set properties

Invoke operations

Can subscribe for notifications (and will update automatically)

Basic Scenario

MAX can be used to connect to any WSDM Resource



- Use EPR to indicate which resource
- Supports Service Groups

