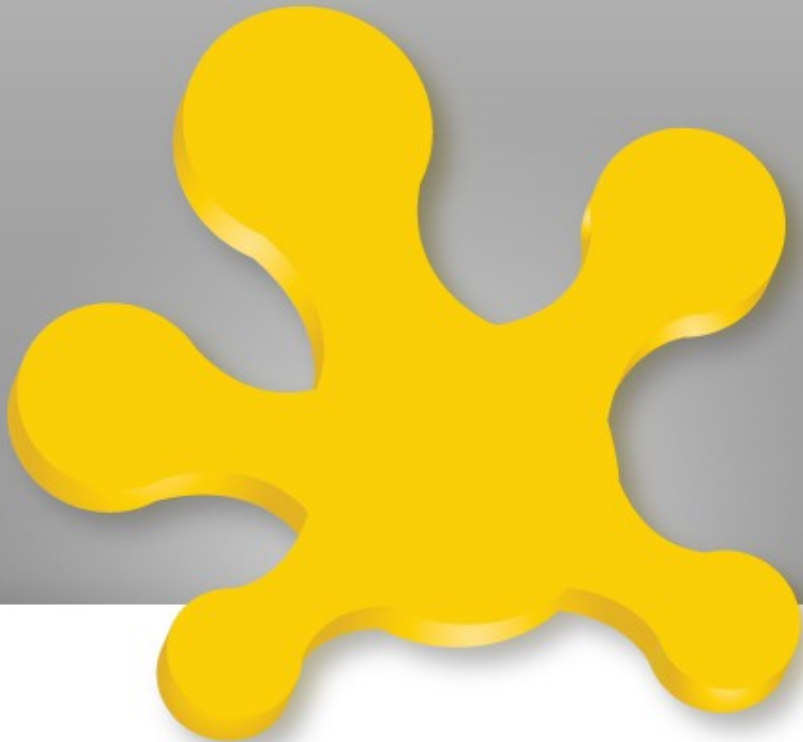


Krzysztof Daniel, Jakub Jurkiewicz

OSGI



Pre OSGI-era

- Developer's view
- VM view



Common problems

- Dependency management
- Class shadowing
- Class-loader hell

Open Service Gateway initiative

- Aims at providing software layer which will be able to load and unload modules without restart
- Services management in cars, telco, intelligent buildings, small offices etc.

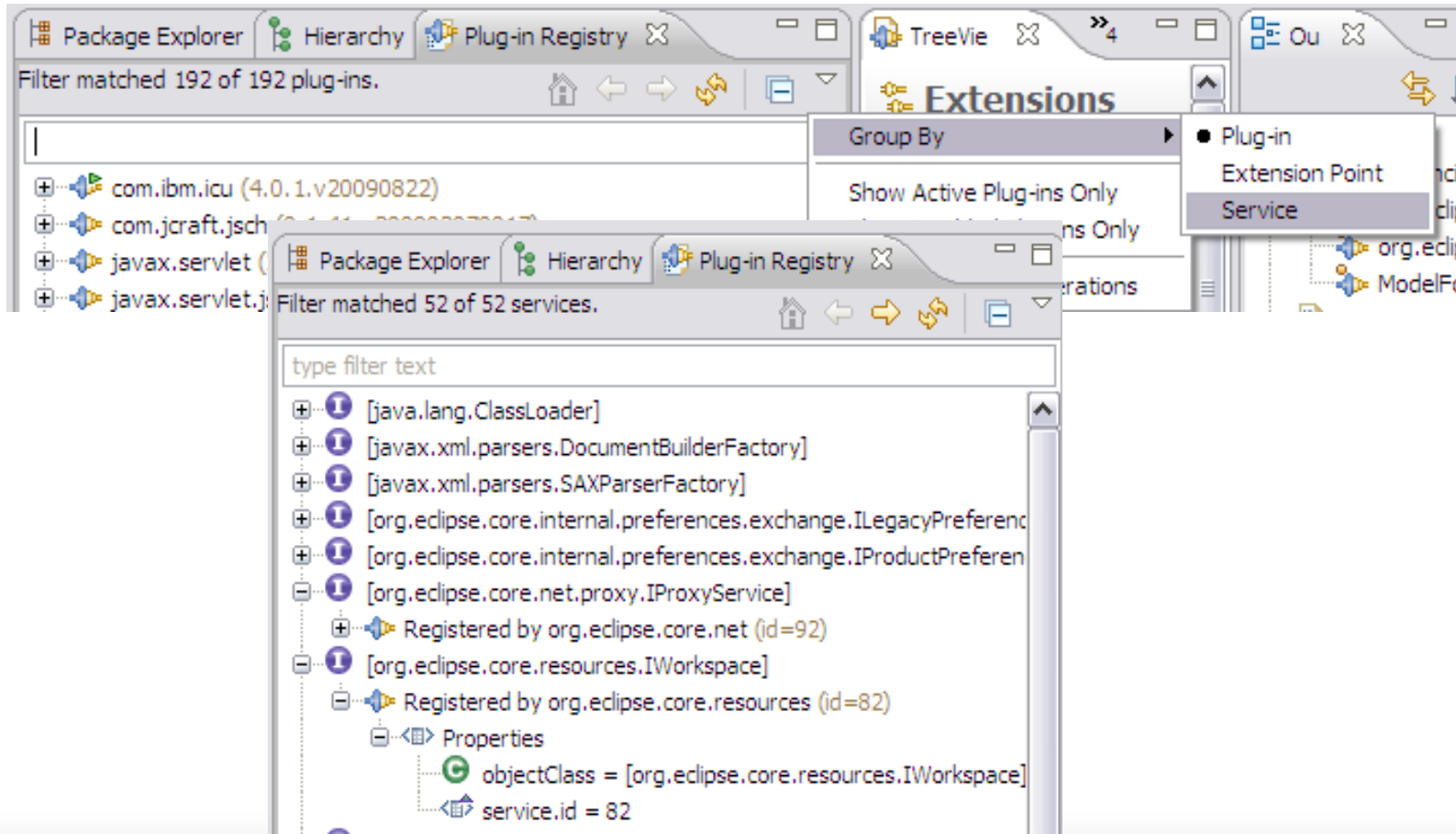
Services

- UPnP
- Logging
- Runtime
- Event admin
- Many many more

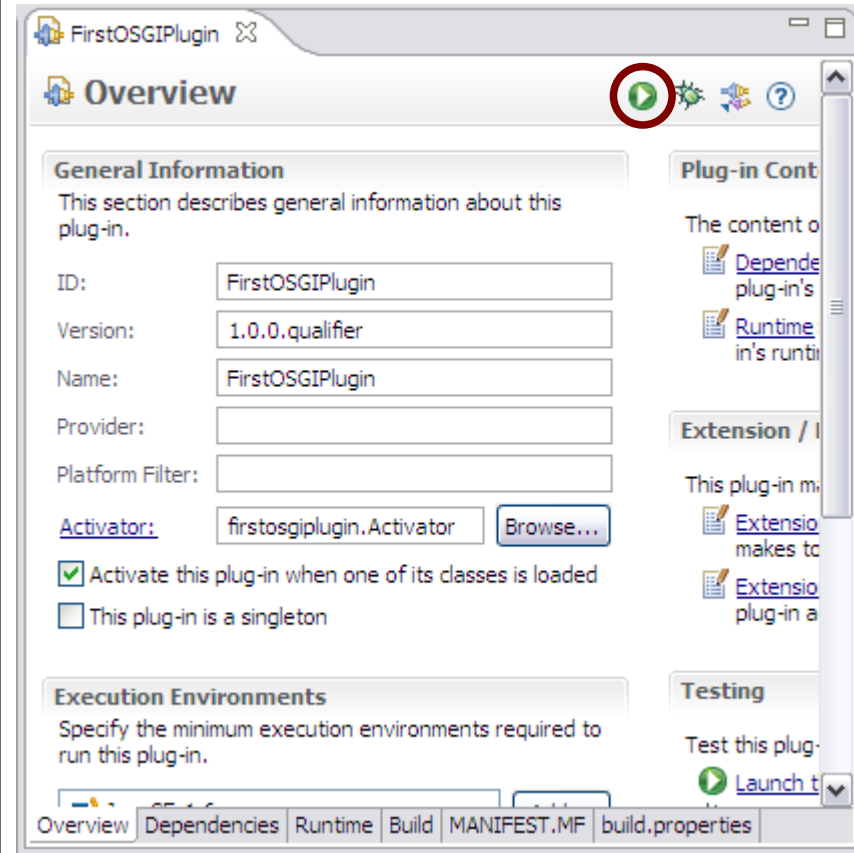
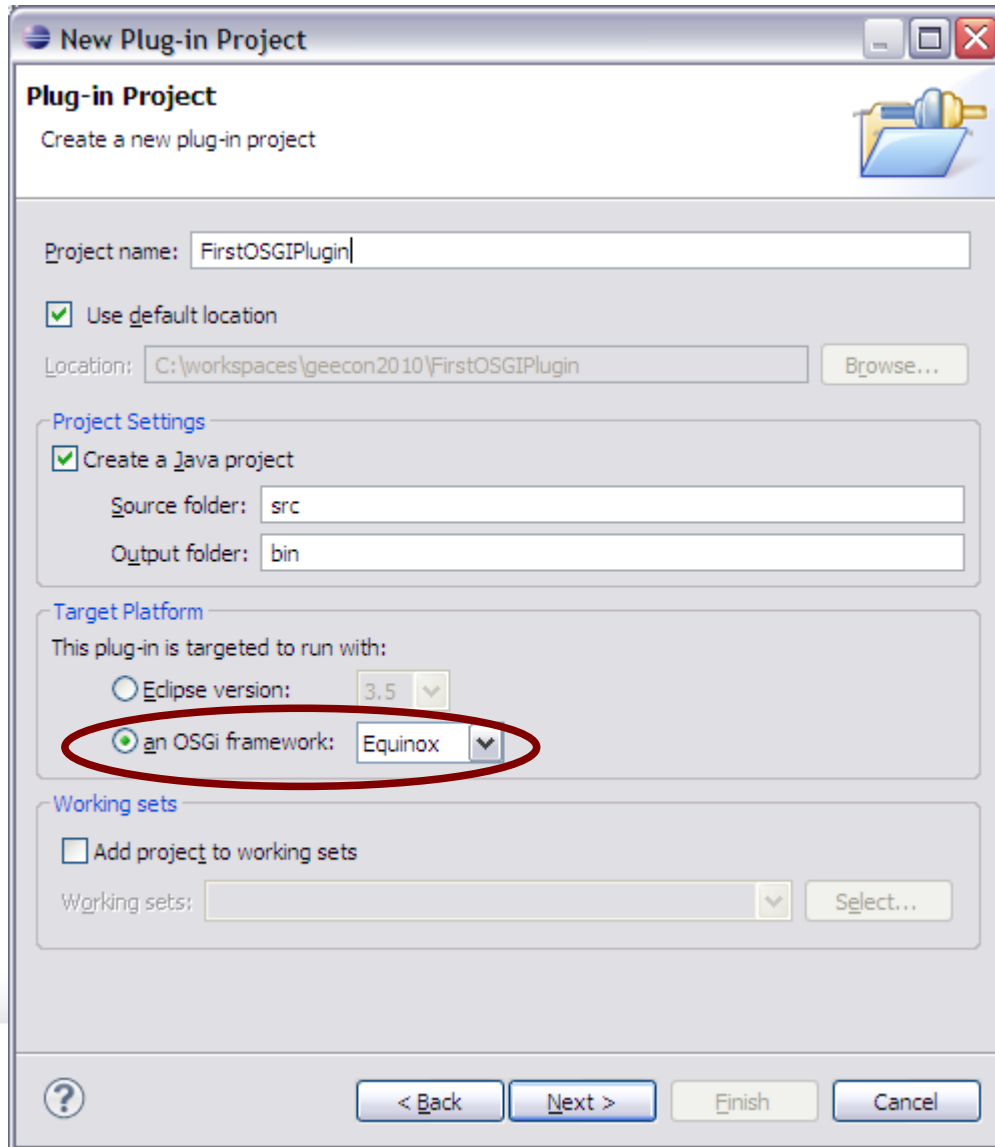
But what about the jar hell?

- extended Manifest.MF
 - required bundles (version)
 - required packages (version)
 - exported packages
 - additional parameters
- everything else is hidden
- Scoped class loaders

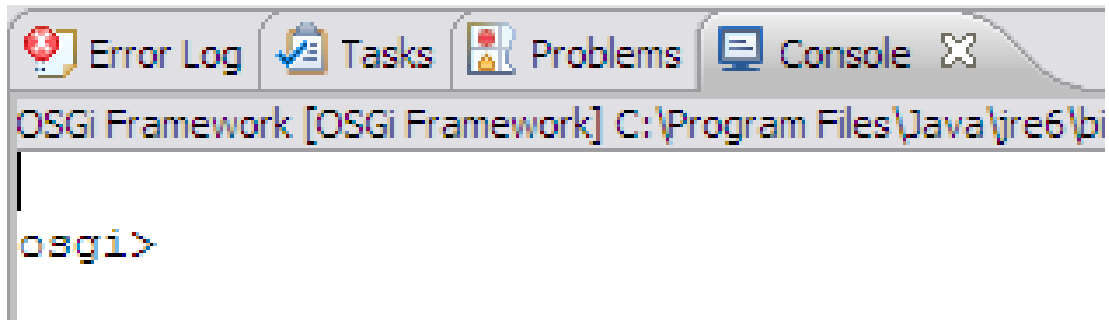
Are there any services in Eclipse?



Your first OSGi plugin



Is that all?



The screenshot shows a console window with a title bar containing tabs for 'Error Log', 'Tasks', 'Problems', and 'Console'. The main area of the console displays the text 'OSGi Framework [OSGi Framework] C:\Program Files\Java\jre6\bin' followed by a vertical line and the prompt 'osgi>'.

- ss
- help

Short Exercise

- create a plug-in
- start & stop it
- observe method execution order
- create an application (non UI „hello world“)
- startApp

Hint: apps are not being run in the main thread

Services

- What does „service” exactly mean?
- Publish – Subscribe
- Declarative services

Declarative Service

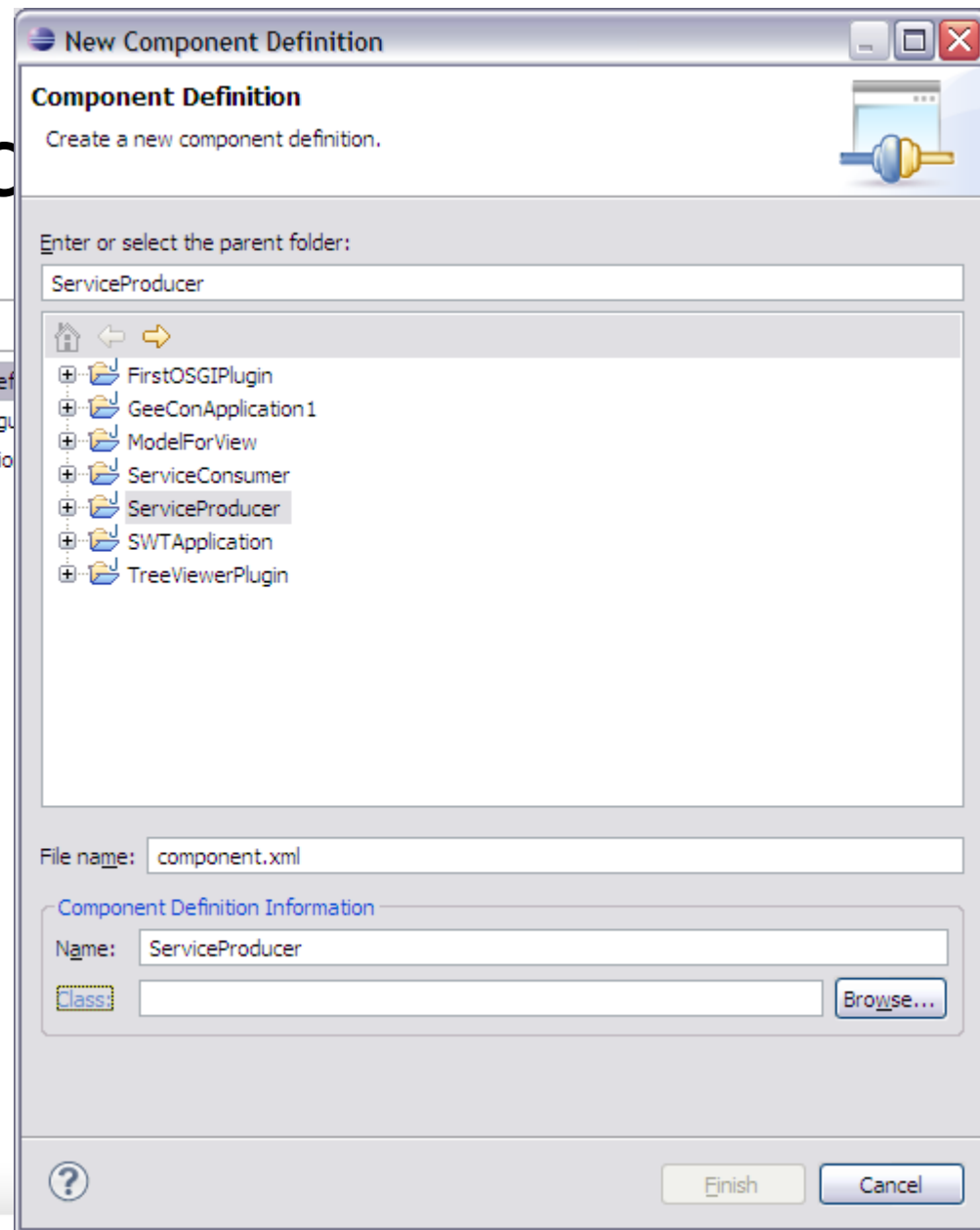
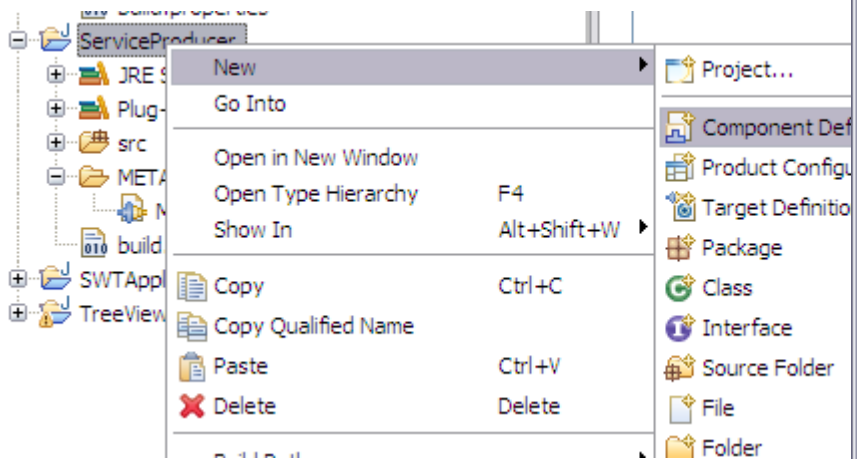
- Automatically managed
- Bound to bundle lifecycle
- Defined in the xml

Your first declarative service

```
public interface ICurrencyConverter {  
    public String getCurrency();  
    public double convert(double value);  
}
```

- Must be exported


Your first dec



Your first declarative service

New Java Class

Java Class

 The use of the default package is discouraged.

Source folder:

Package: (default)

Enclosing type:

Name:

Modifiers: public default private protected
 abstract final static

Superclass:

Interfaces:

Which method stubs would you like to create?


public static void main(String[] args)

Constructors from superclass

Inherited abstract methods

Do you want to add comments? (Configure templates and default value [here](#))

Generate comments













New Component Definition

Component Definition

Create a new component definition.

Enter or select the parent folder:


  

-  FirstOSGIPlugin
-  GeeConApplication1
-  ModelForView
-  ServiceConsumer
-  **ServiceProducer**
-  SWTApplication
-  TreeViewerPlugin

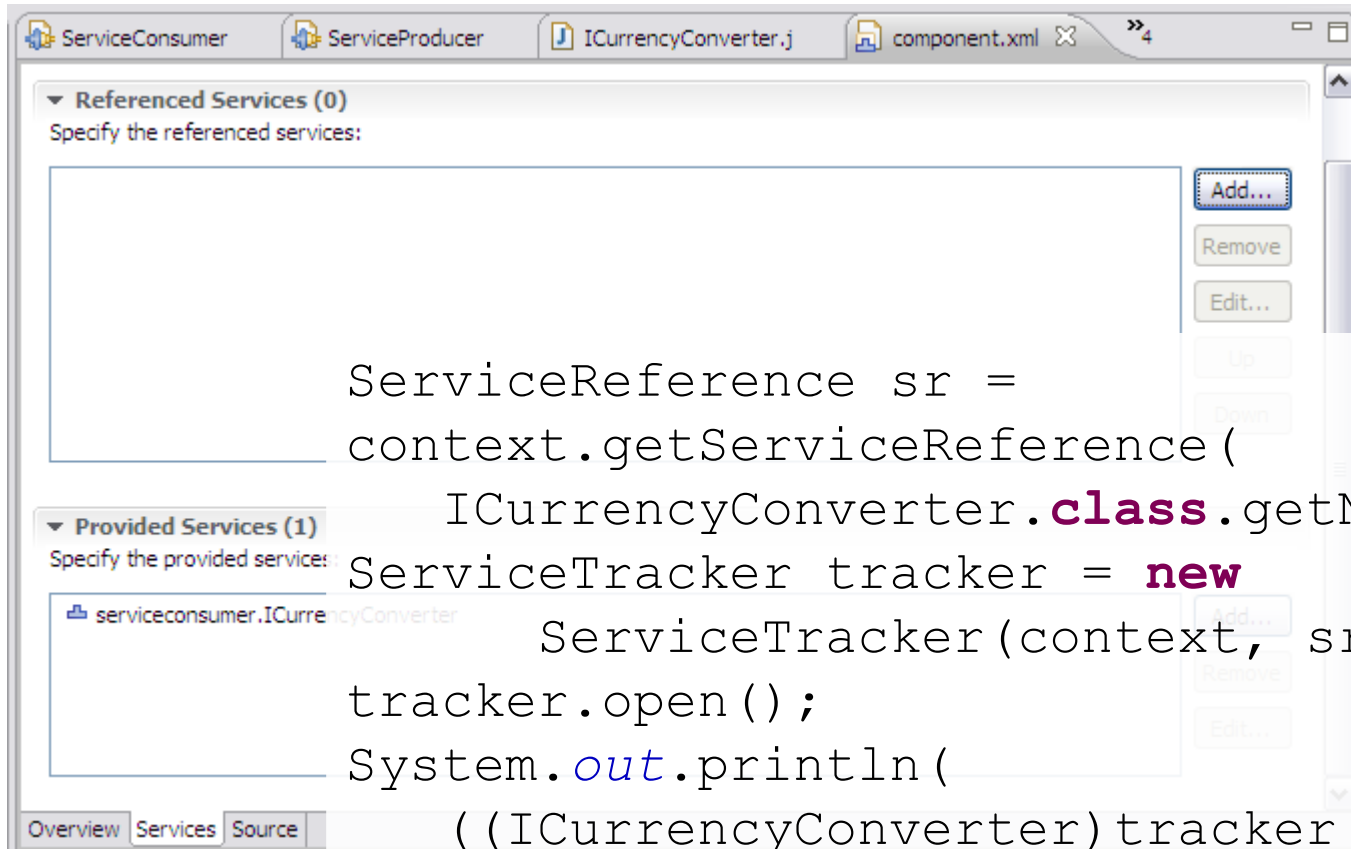
File name:

Component Definition Information

Name:



Your first declarative service



An exercise

- Launch OSGI application
- Start & stop bundles to observe when the service is available

More complex service

The screenshot displays an IDE interface with the following components:

- Overview Panel:** Shows component details for `ServiceConsumer`. The `Class*` field is highlighted with a red box and contains `serviceconsumer.ServiceConsum`.
- Services Panel:** Shows a list of **Referenced Services (1)**. The entry is `0..n ICurrencyConverter [add,remove]`.
- Referenced Service Dialog:** A modal dialog for configuring the service with the following fields:
 - Name:** `ICurrencyConverter`
 - Interface*:** `serviceconsumer.ICurrencyConverter` (with a `Browse...` button)
 - Cardinality:** `0..n`
 - Policy:** `static`
 - Target:** (empty)
 - Bind:** `add`
 - Unbind:** `remove`
- Buttons:** `Add...`, `Remove`, `Edit...`, `Up`, `Down`, `OK`, and `Cancel` are visible.

Final Task

- Workers (cost per hour, reliability) – many plugins with the same service
- The manager – gathers all the workers in one place
- A view – regular Eclipse view which uses the manager service to display the list of workers

Q & A