OSGi:: Wire Admin Service

Pequeña introducción
Alejandro de Celis
Doctorado ESIDE 2005-2007

Remembering OSGi

- Open Services Gateway Initiative
- Small layer
  - Multiple, Java™ based, components to efficiently cooperate in a single Java Virtual Machine (JVM).
- Extensive security model
  - Components can run in a shielded environment.
- The OSGi Framework provides an extensive array of mechanisms to make this cooperation possible and secure.
The OSGi service platform specification has a central feature called a service registry.

It's the place where you register your own services when they are started and where you fetch handles to other services to be able to interact with them.

Usual Interaction Scenario

- Create or get an object representing the service
- Register yourself as a listener to some API
- You get called whenever something happened
  - You will seldom be noticed if this service needs to be unloaded
  - restarted
  - ... or anything else that will affect its availability
- The service is often regarded as a part of the application instead of something separate that could serve several applications at the same time
- These types of applications tend to be more monolithic than applications built upon separated services.
So...?

- The optimal low coupling scenario is when the consumer only knows what type of data it wants to consume, the producer knows what type of data it produces, but neither of them know anything about the other.

Wire Admin Service

- It is a... very flexible way to connect producers and consumers to each other.
- Minimizing coupling between consumers and producers.
- The Wire Admin Service is the “switchboard” with wires, which determines which consumer will be connected to which producer.
Wire Admin Service: The Insides (I)

- The consumer implements a Consumer interface:

```java
public interface Consumer {
    public void producersConnected(Wire[] wires);
    public void updated(Wire wire, Object value);
}
```

Wire Admin Service: The Insides (II)

- And the producer implements another interface:

```java
public interface Producer {
    public void consumersConnected(Wire[] wires);
    public Object polled(Wire wire);
}
```
Wire Admin Service: The Insides (III)

The Wire Admin Service Interface:

```java
public interface WireAdmin {
    public Wire createWire(String producerPID, String consumerPID, Dictionary properties);
    public void deleteWire(Wire wire);
    public Wire[] getWires(String filter) throws InvalidSyntaxException;
    public void updateWire(Wire wire, Dictionary properties);
}
```

Wire Admin Service: The Insides (IV)

To create a wire between the navigation application and the position producer, a call to the WireAdmins `createWire()` method is made with the consumer and producer PIDs as arguments.

This will create a Wire object, which represents the link between them, but also is what gives them possibility to communicate and still being totally decoupled from each other.
Wire Admin Service: The Insides (IV)

public interface Wire {
    public Class[] getFlavors();
    public Object getLastValue();
    public Dictionary getProperties();
    public String[] getScope();
    public boolean hasScope(String name);
    public boolean isConnected();
    public boolean isValid();
    public Object poll();
    public void update(Object value);
}

Wire Admin Service: The Insides (V)

- The `consumersConnected()` and `producersConnected()` methods on the Consumer and Producer interfaces are called when either service is registered and whenever a new connection is created or lost.
- Both the consumer and the producer will always know its current set of wires.
- When the Producer has a new value, it calls the `update` method on each Wire. Each Wire object calls the `updated` method on its connected Consumer.
Wire Admin Service: The Insides (VI)

- If a Consumer needs to poll for data...
  - Merely calls the poll-method on the Wire
  - The Wire object calls the poll-method on the Producer
  - Data is returned to the Consumer.

Conclusion

- Consumers and the producers have been **completely decoupled**
- A very flexible mechanism has been created
  - Connections can easily changed
    - Even in runtime if needed.
- Consumers can poll for values
- The Wire Admin provides a mechanism to create filters to get data
  - each second
  - when the value passed a certain threshold,
- which is a very useful way to handle data that changes very rapidly without suffering too much performance.
Bibliography

- OSGi Technology: http://www.osgi.org/osgi_technology/
- Choonhwa Lee, David Nordstedt, Sumi Helal. Enabling Smart Spaces with OSGi. PERVASIVE COMPUTING. July-September 2003 (Vol. 2, No. 3) pp. 89-94
- Niclas Nilsson BLOG: Connecting Producers and Consumers http://niclasnilsson.se/articles/tag/soa