Facade Design

by Paul Hammant

1. Introduction

AltRMI publishes an object via its interface. It does not replicate the object on the client side, it generates proxies for it defined as facades.

2. Facades

Consider a system that models weather stations (fixed and mobile) and the meterologists that staff the stations.....

Picture of Facades

The interfaces and the class on the right of the green line are 'interface' or API, and we want them to exist as is on the client side for general use. Things on the left are the implementation classes and they exist on the server side only. Though not shown here, it would be easiest to have them in a separate package. Representing those objects on the client side are generated proxies. Proxies are pass-by-reference boundaries but are castable to any of the interfaces they represent. There is one pass-by-value object and that is Coordinate. It should be serializable and final (Immutable pattern).

The principle point of entry into the system from the client point of view is 'WeatherSystem'. The mechanism of entry is a lookup on an agreed name. We recommend 'WeatherSystem' or 'WeatherSystem_1.0' etc.

Once the client has a handle on the WeatherSystem normal Java tarversals are possible

```
WeatherSystem ws = getWeatherSystem(); // some thing that does the JNDI lookup.
// yes we know the following could throw NPEs or Array Index issues.
String aName = ws.getWeatherStation("ArcticOne").getMeteorologists()[0].getName();
```

To generate the correct proxies for the above, you would want to have the interface as 'WeatherSystem' and additional-facades of 'Meteorologist' and 'WeatherStation'

Copyright (c) @year@ The Apache Incubator Project. All rights reserved. $Revision: 1.2 $ $Date: 2003/02/16 21:41:35 $