HiveMind

HiveMind is a services and configuration microkernel. Details about it are available at its home page and on Howard's Tapestry and HiveMind Blog.

It will be used as part of Tapestry4 to form the infrastructure of Tapestry. Some aspects of the Tapestry DTD will be deprecated, such as <service> and <extension>. Tapestry will define extension points to accomplish the same thing.

In addition, new elements, such as <service-binding> will be added, to easily connect HiveMind services to component parameters.

About the new elements. Is a new binding type needed? I mean, you just need a handle to the service to bind it.

I'm thinking a variation of <property> would be nice..

```
<service-property name="boo" service-id="my.module.MyService" interface="com.x.y.MyService"/>
```

Would be cool if the class enhancement worked for this as it does for <property>

i.e.

```
public abstract MyPage extends BasePage {
  abstract MyService getBoo();
}
```

GeoffLongman

Excellent idea Geoff. We could even make the interface attribute optional (that can be determined at runtime). But this is **certainly** simpler than introducing <set-service>, <bind-service>, etc., etc.

HowardLewisShip

I hope to discover other things like this as I continue integrating Hivemind services into our Tapestry 3.0 application (doing so is slightly cumberson right now!). Am I the only one doing this right now?

To complete my first thought. The idea for `<service-property>` arose from the fact that we needed to have access to the service both in the java code and in ognl expressions in all the places that such expression are useful.

I'm also toying with the idea of a session local service. Although, progress is slow as I learn how interceptors and schemas work. What I'm shooting for now is an interceptor that uses some session aware helper services to persist/restore the state of some pooled services that are not explicity session aware. Thus it could be possible to make any service session local by adding the interceptor. Not there yet, and its not rock solid production quality code now but at least I'm visiting the issues.

GeoffLongman

Or, one could ditch changes to the DTD all together and synthesize a page property ('hive' below) that can be used to lookup services..

```
<property
    name="boo"
    initial-value='hive["my.module.MyService"]'/>
```

GeoffLongman

HowardLewisShip: The <inject> element works great. It's better than the hive option (which is similar to what I came up with for TheServerSide); <inject> is much more efficient at runtime. An initial-value is re-evaluated when the component is constructed, and at the end of each request cycle. The implementation of <inject> evaluates the HiveMind object reference once, when the page or component is being enhanced (see ComponentClassEnhacement), and retains that value, passing it into the page via a constructor, and storing it in an ordinary instance variable.