

StrutsState

Struts 2.0 and [StrutsJerichoState](#): an addendum to [StrutsJericho](#)

Let us consider \implies and \impliedby and \iff to be bivalent operators such that $A \iff B$ means that A is somehow bound to B and B is somehow bound to A.

The MVC design architecture (not pattern) in Struts is, let us say, "webMVC", which will mean that contiguous and contiguous parts of the architecture alone are bound together, or:

```
VIEW  $\iff$  CONTROLLER  $\iff$  MODEL  $\iff$  MODEL DATA
```

This "picture" considers the "VIEW" to be JSP, Velocity, etc. rather than the HTML, which is the real "VIEW".

One of the difficulties with writing web apps is that the app storage is never quite right, i.e. the scopes request, page, session, and application do not fit app needs. What is more, lacking any reasonable facade for these scopes, there is a lot of extremely repetitive hard coding relating to them that makes taglibs over complicated and workflows rather mysterious. What is needed, I suspect, is a storage unit for app state relating to the view. Something like:

```
VIEW  $\iff$  STATE  $\iff$  MODEL
```

"STATE" here are classes which are accessible from the VIEW and communicate with the MODEL. In this architecture, SCOPES, e.g. request, page, session, application, should be considered to be an element, like VIEW, CONTROLLER, MODEL, MODEL DATA and STATE. Likewise, the HTML should be considered to be "one with" (aum) the RESPONSE. So, "VIEW", remember, is the dynamic view app, such as JSP or Velocity.

Below is a snapshot of what I have in mind. I have been doing some things like this for a while.

If anyone else is interested, I will consider finding a spot we can work together on this.

I have also considered creating a JerichoFaces for the chaining. But, that is later, for now, and much more ambitious! 😊

Dakota Jack

Snapshot

[Snapshot of this Idea](#)

Involved Developers (everyone welcome – this is a working community open to fundamental suggestions)

Dakota Jack

Amanpreet Singh

State Definitions Relating to Requirements

Screen

Screen = df. The interface between the user (person, machine) and the View. For example, the monitor which renders what the browser "sees".

View

View = df. The file or other data that is read to create this "screen". E.g. the HTML file read by the browser.

Client Viewing \iff Screen \iff Client Delivery App \iff View

View Entities

View Entities = df. Files which either not the View source (HTML for a browser) but which when read "render" the View source, for example JSP, Velocity, etc. Also called "Dynamic View Entities".

View \iff Server Side State Creation \iff View Entities

\implies View Entity Renderers \implies

View Entity Renderers = df. Code in the View Entities that renders the View and which relates either to the "look" of the View or to resources on the client or server side.

View \iff View Entity Renderers \iff View Entities

Look Resources

Look Renderer = df. Code which renders markup language alone in the View.

View Resources

Resource Renderer = df. Code which renders code in the View that accesses data or applications on the client or server. Flash and [JavaScript](#) plugins are covered by this as well as streamed data.

Distinguish between Model Data and View Data (State)

View Data = df. Data that is presently being accessed (streamed), being held (viewed, cached) by the View, or that is on the server side but is relating to processes and to actions which the view (end user) is presently undertaking. Model Data = df. Data store.

State Assumption

State Assumption = df. Something in the Screen which assumes a data state which may or may not exist at the time a request is made. (Necessary to define and to keep in the processing algorithms relating to state management. E.g. this situation should be "earmarked" and then checked on a request. If the assumed state has changed, then the screen should reject the request (if relevant to the state change) and both inform the user of the change and provide a new screen reflecting the change. This is related to but different from notifications that requests were not successfully processed.