

Hourly Student Appointment (HSA) Web Dynpro ABAP review

Larissa Kushkuley, Mark Prudden, Kevin Lyons

April 26 and 30, 2012

Overall:

- Used single, high level Components for most applications with back-end logic in an Assistance Class (not the MV and VC method of SEMO and EL)
- Tried to create WDA Components at a level that “fits”
- Decision on putting more code in the Component Controller versus Assistance Class
 - Component Controller code is available to all views within Component
 - Component Controller code, when making a copy/clone, comes over in-tact
 - SAP recommends putting most non-UI logic into Assistance class for performance reasons – no time to test, no issues encountered to date
- Be careful of using WDDOINIT method. Only call on 1st instantiation of class, not every time screen is called.
- Roughly every web page corresponds to a WDA View.
- Could probably break Views down into smaller components, often realized after-the-fact.
 - [Mark] I would have said could “possibly” break views into smaller components. This is another area for discovery. It may or may not prove to be a better approach.
- Question: when to put things in Windows versus Views (UI logic)
- Naming conventions:
 - ZUTALORS already catches most things.
 - Some of the SAP conventions rather lengthy, could be shortened if agreed.
- Search Help component worked well as a reusable component for HSA
 - However, no way to default Enter focus
 - [Mark] The “however” portion makes it sound like this is a negative side effect of the approach used, which it is not. The focus issue would exist regardless of the way we provided search helps and was merely one more thing that I never had time to investigate. It should become another item on the “discovery” list.

Discovery question:

- Decide UI precision up-front, if a high degree of flexibility needed, go with Java
- Example: Jagged right-edge of input fields, build development effort into scope (not straightforward!)
 - [Mark] I don’t think I agree with this example. The right-alignment of fields was indeed a pain, but it is no reason to go to Java. I would argue that we simply shouldn’t waste time on minor cosmetic issues.

- [Mark] I think it was Larissa's contention that a highly dynamic application may make us consider using Java (Larissa, please correct me if I'm wrong). Personally, I would argue that we should use simpler designs that employ delivered Web Dynpro functionality such as making input fields read-only when appropriate instead of dynamically replacing them with text views (as noted immediately below).
- [Larissa] To clarify the criteria of choosing WebDynpro versus Java for a project (if possible); and if WebDynpro is chosen – agree upfront, as Mark said, to "use simpler designs that employ delivered Web Dynpro functionality", disregarding possible minor display issues;

3 major pain points – huge time consumers:

1. Gyration from Input Fields to Read-only Fields based on user authorizations.

[Mark] (In our application, the switching wasn't normally based on authorizations, it usually occurred when a dropdown list only had one possible value.)

- Grid layout could bind things to background (can't do in Matrix layout) but Grid has other spacing issues

2. Appearance (On/Off) of certain fields based on user authorizations.

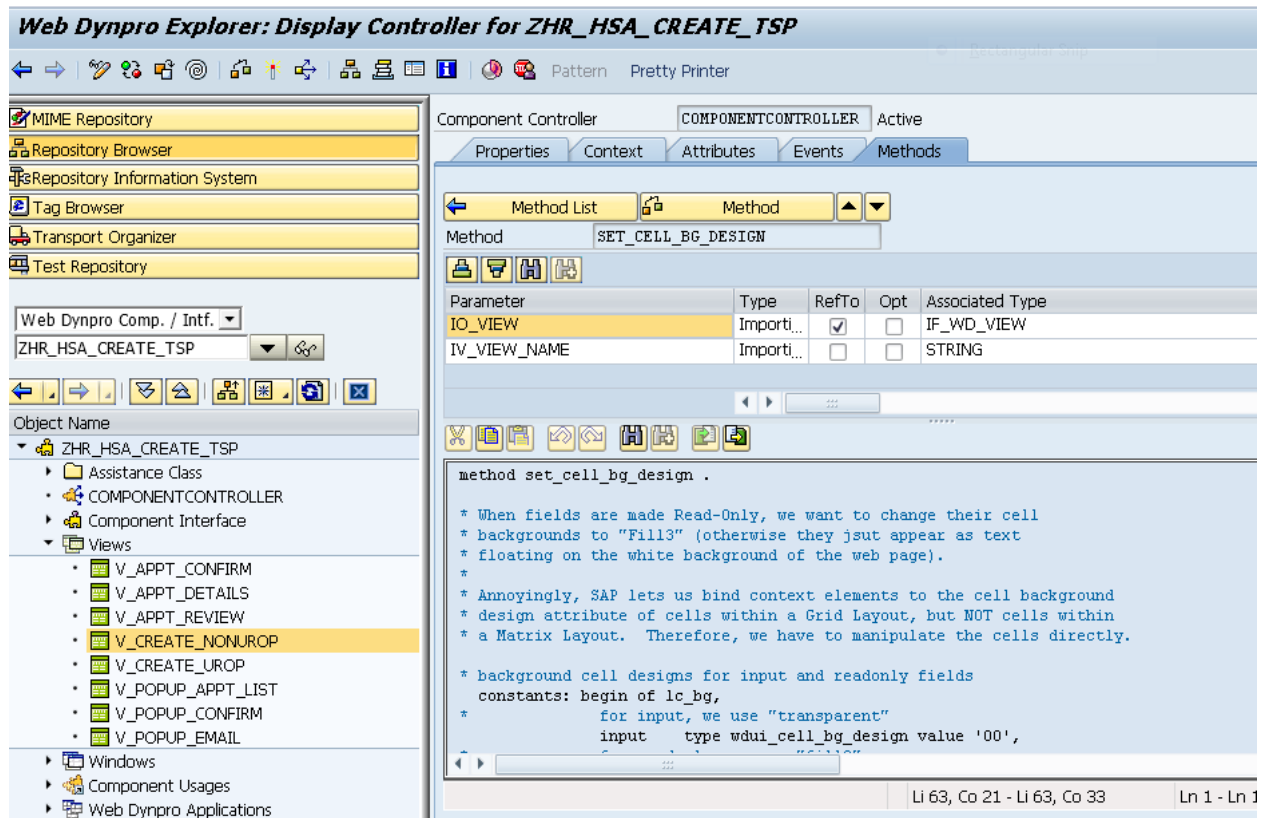
[Mark] (In most cases, I would imagine this showing/hiding is not based on authorizations, but on the "type" or "flavor" of object being manipulated. For example, different fields apply depending on the type of appointment, work order, address, etc. being created/changed/displayed.)

- Leaves behind certain trace when suppressed
- Could use a view container, but causes Accessibility problems
- [Mark] I don't recall this point (perhaps it was yours or Larissa's?). I don't know what the accessibility problems are.
- Should be a better way to Show/Hide – View Container or Layout (Grid/Matrix) comparison

3. Align edges

- Far too much time spent on aligning edges of fields with different lengths
- [Larissa] To add to alignment issues: personally I had a lot of problems aligning tablefield (student info on top of the page) with simple fields (like position details); I would appreciate to have a little more clear guidelines of what widths to use (where/when to use pixels, or other measures like em/ex; when to specify Properties-width vs Layout-width etc),

One method SET_CELL_BG_DESIGN in Component Controller to control all backgrounds for read-only and display as text:



For table displays, did not use Web Dynpro ALV. [Mark] (Not sure if we would have used ALV or not, had we had the time to investigate. This could be another discovery item.)

However did code for table sorting by columns via Actions within the View: [Mark] (and this was really quite easy to add)

ransactions

Sort — had to code logic (not ALV)

on #	Type	Name	MIT ID	Submitted Date	Effective Date
1	Change	Sumit Dutta	970593437	04/12/2012	04/01/2012
9	Change	Samantha Lewis	910100932	04/24/2012	09/06/2011
5	Create	Qusqiat Uffuq	900004358	02/20/2012	02/20/2012
1	Create	Qusqiat Uffuq	900004358	02/16/2012	02/01/2012
0	Create	Qusqiat Uffuq	900004358	02/16/2012	02/01/2012

Within the Table UI element, assign property OnSort to an Action:

View

V_SEARCH

Active

Properties

Layout

Inbound Plugs

Outbound Plugs

Context

Attributes

Actions

Methods

ViewContainerUIElement: VC_HEADER

MessageArea: MA_MESSAGES

V_SEARCH.PAGE.HEADER

Enter the search criteria for the transaction(s) you want to f

You can enter a single value or a range of values.

Transaction #

V_SEARCH.SEARCH_C

Transaction Type

Create Appointment

Student MIT ID

V_SEARCH.SEARCH_C

Preparer Kerb ID

V_SEARCH.SEARCH_C

V_SEARCH

Department

Status

Processed automatically, no action rec

Effective Date

V_SEARCH.SEARCH_C

to

V_SEA

Submission Date

V_SEARCH.SEARCH_C

to

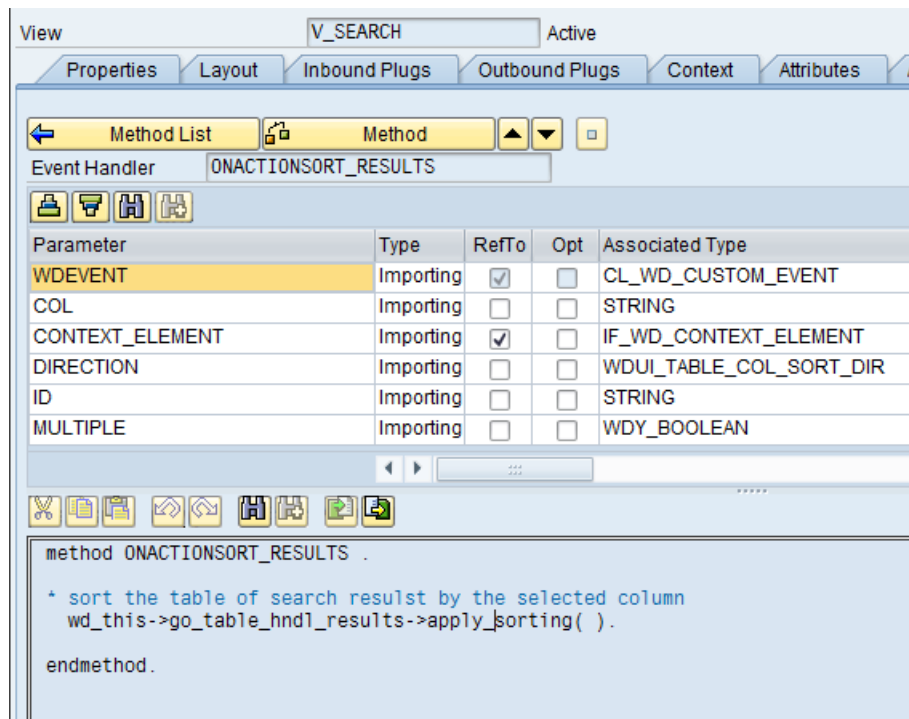
V_SEA

Search

Transaction #	Student Na
V_SEARCH.RESULTS_LIST.REQUEST_NUM	V_SEARCH
V_SEARCH.RESULTS_LIST.REQUEST_NUM	V_SEARCH
V_SEARCH.RESULTS_LIST.REQUEST_NUM	V_SEARCH
V_SEARCH.RESULTS_LIST.REQUEST_NUM	V_SEARCH
V_SEARCH.RESULTS_LIST.REQUEST_NUM	V_SEARCH
V_SEARCH.RESULTS_LIST.REQUEST_NUM	V_SEARCH

- TC_LEFTLINKS
- TC_RIGHTLINKS
- MA_MESSAGES
- FTV_HEADER
- TC_CONTENT
- IE_SPACER3
- BTN_SEARCH
- IE_SPACER4
- TBL_RESULTS
 - TBLC_REQUEST_NUM
 - LA_REQUEST_NUM
 - CAPTION_REQUEST_NUM [Header]
 - TBLC_STUDENT_NAME
 - TV_STUDENT_NAME
 - CAPTION_STUDENT_NAME [Heade
 - TBLC_MITID
 - TBLC_ACTION
 - TBLC_EFFECTIVE_DT
 - TBLC_INITIATED_DT

Property	Value
visibleRowCount	500
width	
Events	
onColSelect	
onCustomFilter	
onDrop	
onFilter	
onLeadSelect	
onScroll	
onSelect	
onSort	SORT_RESULTS
Layout Data (MatrixHeadData)	
cellBackgroundDesign	transparent



For top of page navigation links, each were coded individually per page. For Left-side links ('Go Back', 'Create new appointment') and Right-side links ('Help', 'Close') are specific to each page, so this makes sense.

Pop-up screens are created for warnings. They are asynchronous events in WDA, an Event must be raised when button is clicked on you go to next screen via an Action.

Messaging: some investigation could be done on Error messages. How is WDA behavior similar or different than SAP GUI behavior? For Errors? For Warnings?

[Mark] The HSA project introduced a second level of warnings (a "strong" warning). This complicated things and should never be done again. There should be no reason to ever deviate from the standard of error, warning and success messages.

Texts: all are stored in OTR:ZHR0. It is the recommended approach and gives you flexibility in capitalizing/altering field labels (versus using the default Data Dictionary labels).

Layout types:

- When all read-only fields, matrix layout is good.
- When a mix, difficult because of varying field length, jagged edges.

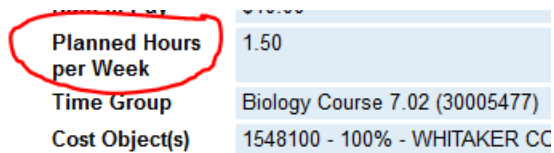
Static attributes (constants). Where is the best place to store these?

Drop down lists – used two primary methods – are there others?

1. From Domain values in Data Dictionary
2. Build in the code of the Component Controller method.

Other smaller UI pain points (easy in HTML, harder in WDA):

Label text that you want to wrap to two lines. Hard coded Width property to force two line wrap.



A screenshot of a UI form with a light blue background. The form contains three rows of labels and values. The first row has the label 'Planned Hours per Week' circled in red and the value '1.50'. The second row has the label 'Time Group' and the value 'Biology Course 7.02 (30005477)'. The third row has the label 'Cost Object(s)' and the value '1548100 - 100% - WHITAKER CC'.

Planned Hours per Week	1.50
Time Group	Biology Course 7.02 (30005477)
Cost Object(s)	1548100 - 100% - WHITAKER CC

Two different font sizes within the same label field. Easy to do by making two label fields, but bad for accessibility (can only map one label to one input field).