

Some thoughts on the Xerte interface

Hello Julian,

To keep everything in one place I will put my comments in this document rather than in the email.

I have downloaded your files and the demo and taken a look at it. I have also read your additional email which asks

“Are there a set number of interface “states” that we can define?”

I think there are two situations when the Author will want to modify the interface,

1. Once, in the navigation attribute of the learning object when they set the default navigation for the site.
2. At various times in the project, from a connector page either,
 - a) On opening the Connector page e.g. to disable some buttons
 - b) On exiting the Connector page to re-enable buttons

For example I could imagine a project that has a Home button, an Historic Back button and a Next Button. As its default navigation. When the project reaches a MCQ Connector page, the page saves the current navigation settings and then all the navigation buttons are disabled. On exiting any one of several alternative routes from the Connector page the user is taken to a sequence of ordinary pages and the Back and Next buttons only are enabled. At the end of each sequence is a Redirector Connector page. On clicking the continue button of the Redirector page the page resets the saved navigation settings and then goes to the destination page which opens with the original navigation thus restored.

This would enable a set number of values to be defined for the learning object navigation attribute. Incidentally a project that uses Connector pages to set up alternative routes through a project may not necessarily open with a menu, so it probably makes sense (at least to me anyway :-)) if there is a new option of “Historic”, rather than to offer the Historic back button for projects using the “Menu” option. The “Menu” option includes the TOC so this may lead to the kind of confusion that you first experienced when you came across an historic back button in a Xerte project.

Possible “states” or values for the Learning Object Navigation attribute.

Key:

A = Active, D = Disabled but visible. An empty box indicates the button is not displayed at all.

	Home	TOC	Back (L)	Back (H)	Progress	Next (Continue)
Linear		A	A		A	A
Menu with Page Controls		A	A			A
Menu		A				
Historic	A			A		A

In the Historic navigation style the Home button should appear without the need to make any further call to the addHomeButton(x,y) method, though having that method available to call will, I

Some thoughts on the Xerte interface

am sure prove useful. Personally I can't see a need to include a Home button in any of the first three navigation options (Linear etc.) as the TOC already provides a route to the first page.

Navigation display settings that may be required as the user works their way through the learning object

I think the most likely options would be

	Home	TOC	Back (L)	Back (H)	Progress	Next (Continue)
Viewing a connector page in a linear project		D	D		A	D
Viewing a connector page in a Menu with page controls project		D	D			D
Viewing a connector page in a menu project		D				
e.g. viewing a connector page in a historic project	D			D		D
Single page of Connector	D			A		D
Sequence of pages off a connector	D			A		A
Possibly useful ?	D			D		A

Possible further “States”

Since in all project types other than “Historic” only have enabled or disabled navigation in the above tables then if a `setInterfaceMethod(stateName)` was used then this could be used to indicate that all visible buttons should be disabled.

The problem comes with the last three “states” we could have

“Back only”

“Back and Next”

“Next only”

But these might only have relevance for projects that started out with “Historic” navigation.

All the above assumes that once the Learning Object's Navigation attribute value has been selected then the only other time you might change the navigation settings is from a connector page. Which raises the question...

What options to offer in Connector Page XWD forms?

We could offer

Combo box value

Enable all

Disable all

Enable Back button only

Enable Back and Next buttons

Enable Next Only

Data

“a”

“d”

“b”

“bn”

“n”

Some thoughts on the Xerte interface

But how would this be implemented in the Connector pages?

“a” (Enable All) must only enable the buttons that are defined in the Learning Object's Navigation attribute to be active. So do we pass the Learning objects Navigation attribute to the `rootIcon.setInterfaceState()` once again. But how would “d” (Disable All) know which buttons to disable?

If the engine stored a copy of the initial settings defined when the Learning Object's Navigation attribute was acted on then you could have a separate method

```
setNavigation(“a”); // enable all
```

That would “know” which buttons to enable, or if “d” was passed to disable or if “b” was passed which back button was being referred to.

So rather than defining a number of additional states to handle the needs of the Connector pages what we need is just the “Historic” state and an additional method that can intelligently use the information provided when `rootIcon.setInterfaceState()` was called.

That would nicely separate the selection of the Learning Objects navigation from the need for temporary settings in the connector pages and provide easy support for handling the temporary settings in the Connector pages scripts.

What do you think?

Kind regards

Johnathan