

Mini OpenMRS Workshop Nov 15th 2007

In Attendance

- Darius
- Ben
- Brian
- Burke
- Paul

Implementing the Provider-pattern in OpenMRS Modules

We need to have a way to register providers of classes in modules.

Should we use extension points?

- a provider will need to register a class
- a provider will need to give a keyword as to what its providing services for
- A provider needs to conform to extend/implement a certain class. Extension points do not currently do this.

Examples of service providers:

- Reports
 - web report provider
 - report provider
- Forms
 - form provider

Testing Sync

- How to set up the two databases?
- Start using sync and start writing unit tests
 - Write some sanity checking functions:
 - Count the number of encounters and obs before and after a sync transaction to see if they are right
 - Keep a log of everything, never delete a sync packet, etc
 - Dump and zip a backup file before every sync transaction?
- Use dbunit and the new testing framework. Very straightforward
 1. A blank database and tables are created by the dbunit framework using the hbm mappings files
 2. Insert known values into the tables
 3. Do some stuff that will be journaled by sync. (Add patient, add names, add obs, run hl7, etc)
 4. Call the sync method to get the "offline" transport file
 5. Blank the database back to the "known" values in the tables
 6. Run the sync on the flat file that was from the other table
 7. Verify that things look as they should compared with what was inserted before

Avoiding OpenMRS Stubs

- Currently, the hl7 processor "creates"/"finds" concepts and links them to obs by creating the concept like

```
new Concept(someConceptId)
```

. It should be creating the concept object by calling

```
ConceptService.getConcept(someConceptId)
```

- The result of not calling `getConcept` is that the GUID field is not filled in and hence sync ignores the concept and doesn't journal it
- The hl7 processor shouldn't use `getConcept` repeatedly in its methods because that is quite a slow process.
- The processor just needs to fill in *both* the `conceptId` and the GUID
 - Batch fetch all guides for all concepts in the current hl7 message
 - Still create a concept "stub" but it has both `conceptId` and GUID
- If the synch processor sees an object without a GUID, throw a warning and attempt to fetch the GUID as well

Reporting / Logic

See the google doc http://docs.google.com/View?docID=dgb7h84w_11g45fsf&revision=_latest

Exports by Enc, Obs