FIRST - Bug #3658

Need to be able to save during parsing process

11/12/2008 07:14 PM - Ryan McFall

Status:	Resolved	**Start date:**	11/12/2008
Priority:	Normal	**Due date:**	
Assignee:	Ryan McFall	**% Done:**	0%
Category:	parser	**Estimated time:**	0.00 hour
Target version:	client-prototype	**Spent time:**	0.00 hour
Bugzilla-Id:	3658		

Description

Right now you have to complete the entire process of parsing the exam; otherwise work you have done is lost. This should be implemented.

History

#1 - 05/26/2009 12:30 PM - Ryan McFall

At first I thought java serialization was needed for this. Then I realized that we already have a serialization mechanism for much of the contents of MainWindow: XML.

Right now, the EditQuestionsCommand action in edu.ucsb.nceas.morpho.first.edml.commands has the following code in it:

```
AbstractDataPackage adp = UIController.getInstance().getCurrentAbstractDataPackage();
List assessmentItems = AssessmentItemUtil.getAssessmentItems(adp);
final QuestionList questions = AssessmentItemUtil.dataPackages2QuestionList(assessmentItems);
...
MainWindow.getInstance().getQuestionsPane().setQuestionList(questions);
```

The other information that is necessary in order to be able to restore is:
- the original PDF file that was parsed for this data package
- the set of images that were parsed out of this data package
- the set of pages that should be ignored
- the set of rectangles that should be ignored

(note that the ignored pages/rectangles would strictly be for the user's visual benefit, the parsing process will already be complete, so the program won't use them for anything)

If there was a way that the edit questions command code look to see if a parse had already been started for the assessment in question, and then look up the PDF file and image list when invoking the edit questions command referenced above, restarting the parsing process would be relatively straight forward.

I would propose that the assessment be auto-saved locally before the assessment import wizard process is started. This gives us an assessment ID to use as a sub-directory within the .first directory, so that we can put the images there, as well as storing the original PDF file there after the parsing process is complete. If edit question is subsequently called, the list of images and original PDF file can then be located. Once the data package is saved to the network, the edit questions option may need to disappear. I don't remember if there are versioning issues here; I do know that it won't be very easy for someone on a different machine than the original importer to work with editing the questions, since they may or may not have the original PDF, and certainly won't have any of the extracted images, as these come from the parsing process, and the parser won't be able to come up with the same set of questions/images as the user had after they edited them.

In order to implement this change, the chooseAndImport method of MainWindow will need to be changed to take at least an assessment identifier as a parameter, so that it can make the appropriate subdirectory under the .first directory.

#2 - 05/27/2009 07:33 AM - Ryan McFall

Another issue that complicates this process is that Sandeep is now adding page number and possibly bounding box information to the Question class. This information will not be written as part of the serialized XML stored on the server. We could put this information into the XML if the schema allows for it (might be relevant information anyway). Or, we could serialize this information in a different location from the XML, and then reload it after loading the XML.

#3 - 05/27/2009 10:10 AM - ben leinfelder

I think this is doable, but I'm not sure if it's exactly what we want every time there is a request to "Edit" the assessment.
-I currently like the fact that I can start from scratch with a new and completely different PDF file (say if I realize I've parsed the incorrect exam). If we create too strong a tie to a particular PDF, I think that could be frustrating. Sure, we can have the option to start from scratch versus continue with what we started, but that adds complexity to development and the UI.
-I think the point about editing questions that have been retrieved from the network is non-trivial (someone else parsed and uploaded the exam and now we downloaded it and have the right to make modifications). It basically means that the editing interface looks very different depending on where the exam originates. If that's acceptable, then okay, great. But I don't think we're going to be uploading all the artifacts from the initial parse whenever we save the assessment to the network.

As far as I can tell we want to allow people to see:
-the PDF for reference ("what was that question again?")
-the images so that they can be added to questions etc.
-are the bounding boxes and ignored pages crucial for anything, really?

I think we'd already identified that they PDF should be saved with the assessment metadata (well, a pointer to the PDF in the metadata). I think this would be a more elegant solution than the .first/assessment.id.directory approach - and it would allow network-saved exams to have access to the original PDF.

For images - i guess you do need to save them somewhere in the .first directory and make them available only to folks who originally did the parsing.

There are likely some demonic details (i.e. "devil in the details") regarding revision numbers for assessments and what the temp directory should contain. For example:
-I parse an exam1.pdf file with 3 images. Save. revision=1
-I make a slight change to one of the questions. Save. revision=2
-I change my mind about the PDF file I was using for the question source. Re-parse exam1b.pdf. Save. revision=3
-Question: do I overwrite all the contents in my .first/assessment.id.directory with the new exam1b.pdf artifacts? or do I have individual directories for every assessment.id.revision?

#4 - 05/27/2009 11:59 AM - Ryan McFall

Ben and I talked about this, and we think it's doable. Here are my notes from what we talked about, mostly for my benefit, but also so that Ben can double-check them.

For the client prototype, we will not allow editing of images after the file has been saved to the network. Question text/metadata will be modifiable. The reason to avoid images is simply complexity. It should eventually be doable, but right now we probably can't make this happen.

We will store a pointer to the PDF file in the metadata for the data package, so that it is accessible both locally and from the network. This requires that the Morpho code has access to the PDF file being edited by MainWindow. I'll write methods to get/set the PDF file being viewed by MainWindow.

We will modify the RunAssessmentParseCommand to ensure that the data package is saved before parsing, so that it has a non-temporary identifier. To do this, SaveDialog will be modified so that it doesn't call setVisible(true) as a result of the constructor. This will instead be done by SavePackageCommand.java. This will be done so that the RunAssessmentParseCommand can construct a SaveDialog, set its fields appropriate, and then call the method SaveDialog.executeButton_actionPerformed(ActionEvent), which actually performs the save.

MainWindow will be modified to take an AbstractDataPackage as a parameter to chooseAndImportFile. This allows us to get the Accession number. The images that are generated by the parsing process will then go into the FIRST profile directory, in a folder named by the Accession number (scheme and doc ID only, without revision number). When editing questions, we'll also pass the current data package, so that we can tell whether it is local or network.

RunAssessmentParseCommand will be modified to decide whether there are already some questions; if so, the user will be asked whether they want to replace the existing questions, or edit them. If edit, we'll invoke the EditQuestionsCommand from within RunAssessmentParseCommand.

#5 - 09/21/2009 08:10 AM - Sandeep Namilikonda

This functionality seems to work fine without any evident errors across
multiple assessments that have been parsed, saved, re-opened, edited, and
updated.

So, I am closing this bug.

#6 - 03/27/2013 02:24 PM - Redmine Admin

Original Bugzilla ID was 3658