CAS Viewer Extension For Provenance Tracking of UIMA CAS Content

CAS Viewer Extension For Provenance Tracking of UIMA CAS Content

1. Proposal

To improve the ability to debug and maintain UIMA components, we propose to add the ability to log the updates to the CAS and the Index Repository as follows:

- track which Analysis Engines (AEs) created or modified the feature structures in a CA
- track the operations (add and delete) by each AE to index feature structures

The collected information can be classified as follows:

- Call sequence to AEs
- For each AE, a list of newly created feature structures (FSs) and a list of changes to pre-existing FSs. We will use ***FS Journal*** as the terminology to refer to this information
- For each AE, a list of added, deleted, and modified FSs to/from index repository (if the same FS is deleted and added back to the index, we will classified as "modified"). We will use ***Index Journal*** as the terminology to refer to this information.

The development of this Provenance Tracking of UIMA CAS Content as described in the Wiki is composed of two parts:

- providing APIs in the UIMA framework to support access of journaling information
- visualizing the collected information
  In this documentation, we will focus on the *GUI part of the visualization* from the end-user's perspective.

2. Development Process

The development will be done as a tooling project of Apache UIMA with the participation from the community. Since there are a lot of codes in the CAS Viewer (submitted as a contribution to Apache UIMA) that can be reused, we propose to develop the visualization of CAS/Index Journal as the extension to the CAS Viewer.

3. Proposed User Interface

In the following proposed GUI mockup, we use *deploy/as/MeetingFinderAggregate.xml* from uimaj-example of the UIMA AS package with some modification to its behavior to illustrate the design. This aggregate AE has the following structure:

```
MeetingFinderAggregate
    Collection Reader
    TokenAndSentence AE
    MeetingDetector Aggregate
        RoomNumber AE
        DateTime AE
        Meeting AE
    Cas Consumer
```

We assume that, after running *MeetingFinderAggregate* with an input document, the following basic information is produced:
(1) A list of calls to AEs
(2) Within the call to each AE, a list of new FSs created by this AE and a list of modified FSs
(3) Within the call to each AE, a list of add/delete/modify FS operations to index repository

The issue here is ***how to visualize the above three kinds of basic information to the developers?***

Note that, for the initial implementation, we propose to only preserve the final value for a FS (intermediate values are not kept).

Based on the above example and assumptions, the following shows some screen-shots of the proposed GUI used to visualize the journal information.

3.1 Viewing Changes to CAS

The information about CAS changes is visualized by the FS Journal tab as shown in Figure 3.1.
The sequence of AE calls is showed in the top section of the tab and is organized as a hierarchy (the key string defined in the aggregate descriptor will be used to identify the AE). The number next to the AE's name is the total number of FSs added or modified by the AE. For example, ***Meeting AE (2 FSs)*** means that there are two FSs added or modified by the Meeting AE.

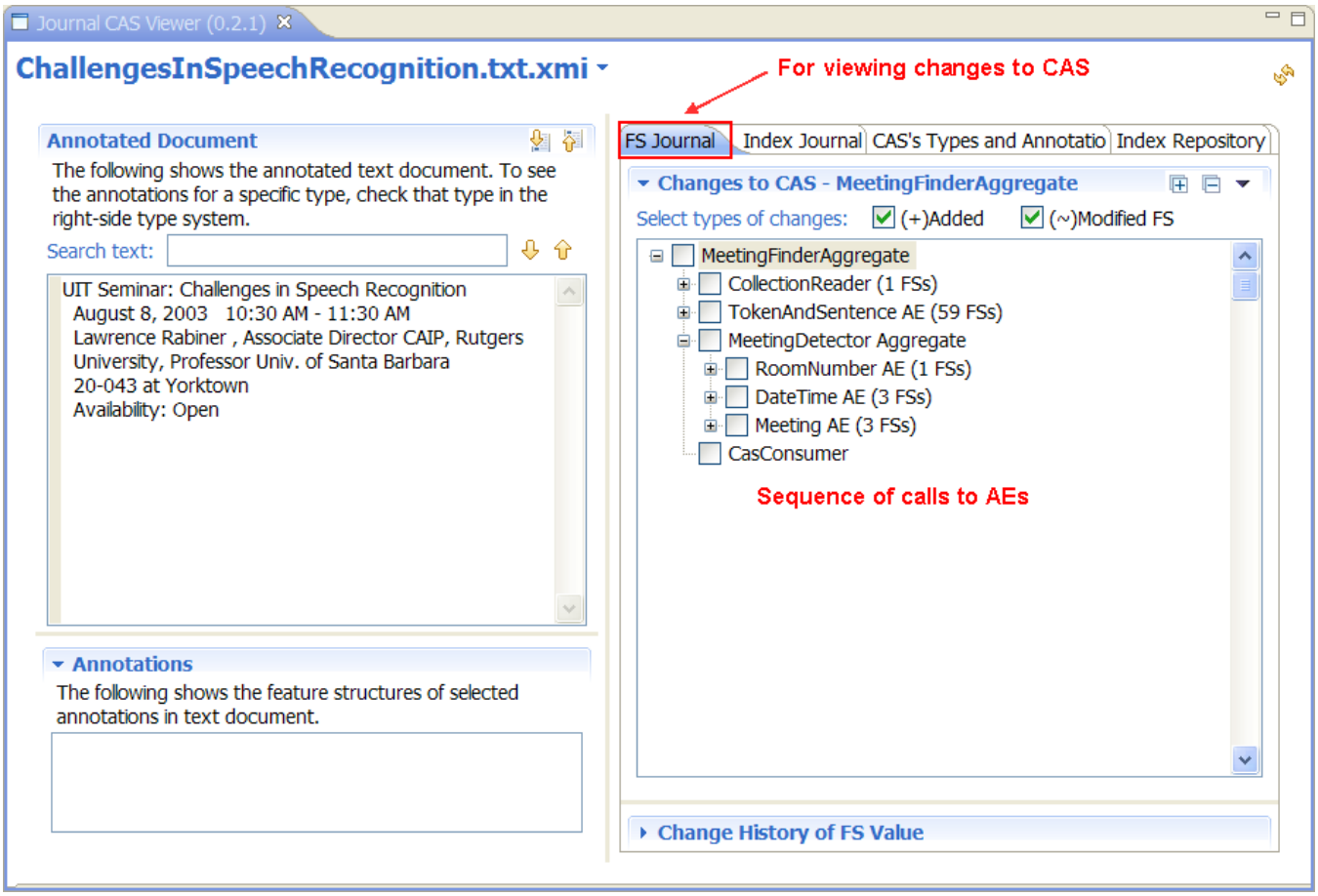

Figure 3.1. For viewing changes to FS in the CAS

Since it is possible to have a long list of FSs (e.g., a few thousands of *Token* annotations), the list of FSs is compressed within the *type name* node and the number at the end of the name indicates the total number of added/modified FSs as shown in Figure 3.2.a.

When the type name's nodes are expanded (by clicking on the + sign), the added and modified FSs are revealed as shown in Figure 3.2.b.

We use the ⊕, ⊖ and (~) signs to represent ***added*** FS, ***deleted*** FS, and ***modified*** FS, respectively. Note that, for the FS Journal, we don't have the case of deleted FSs.

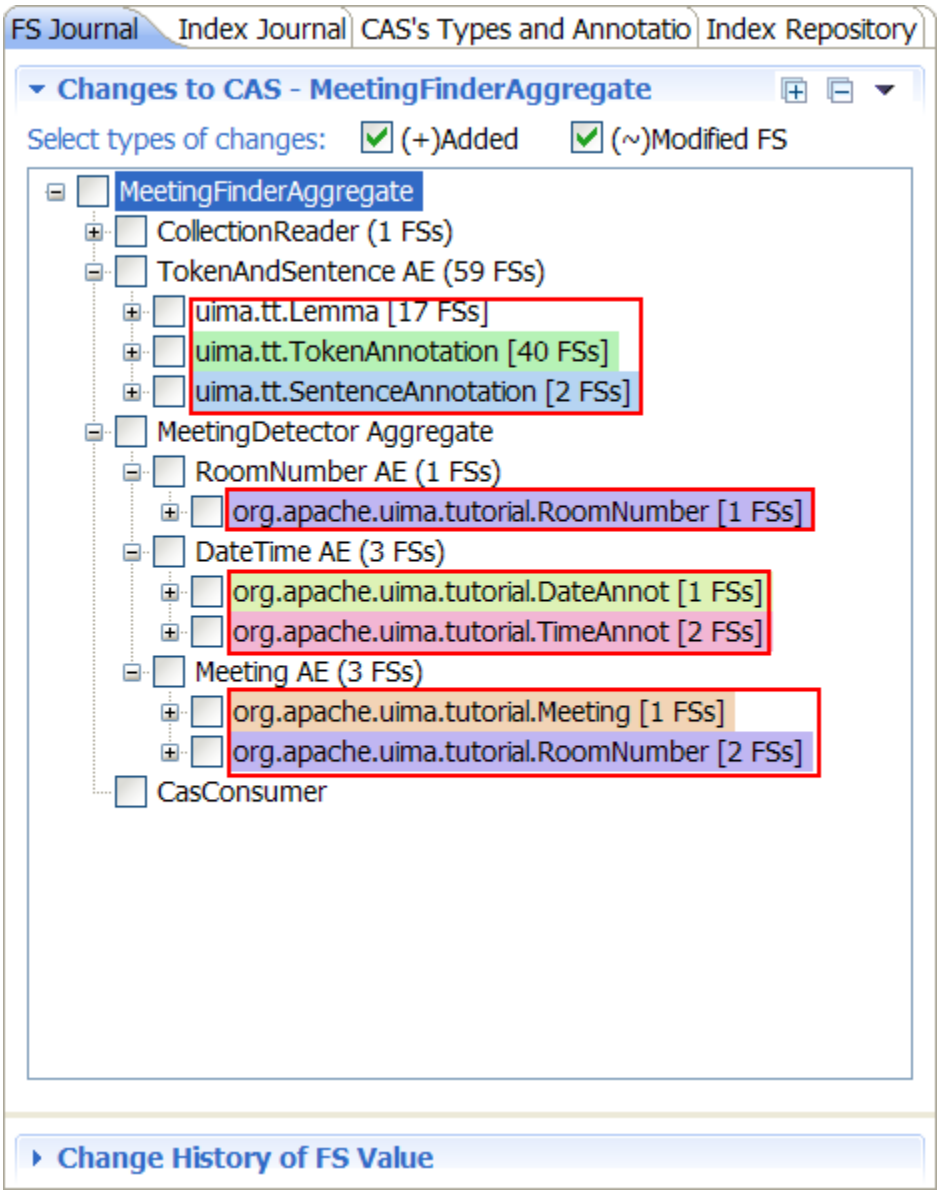

Figure 3.2.a. List of type nodes containing changed FSs

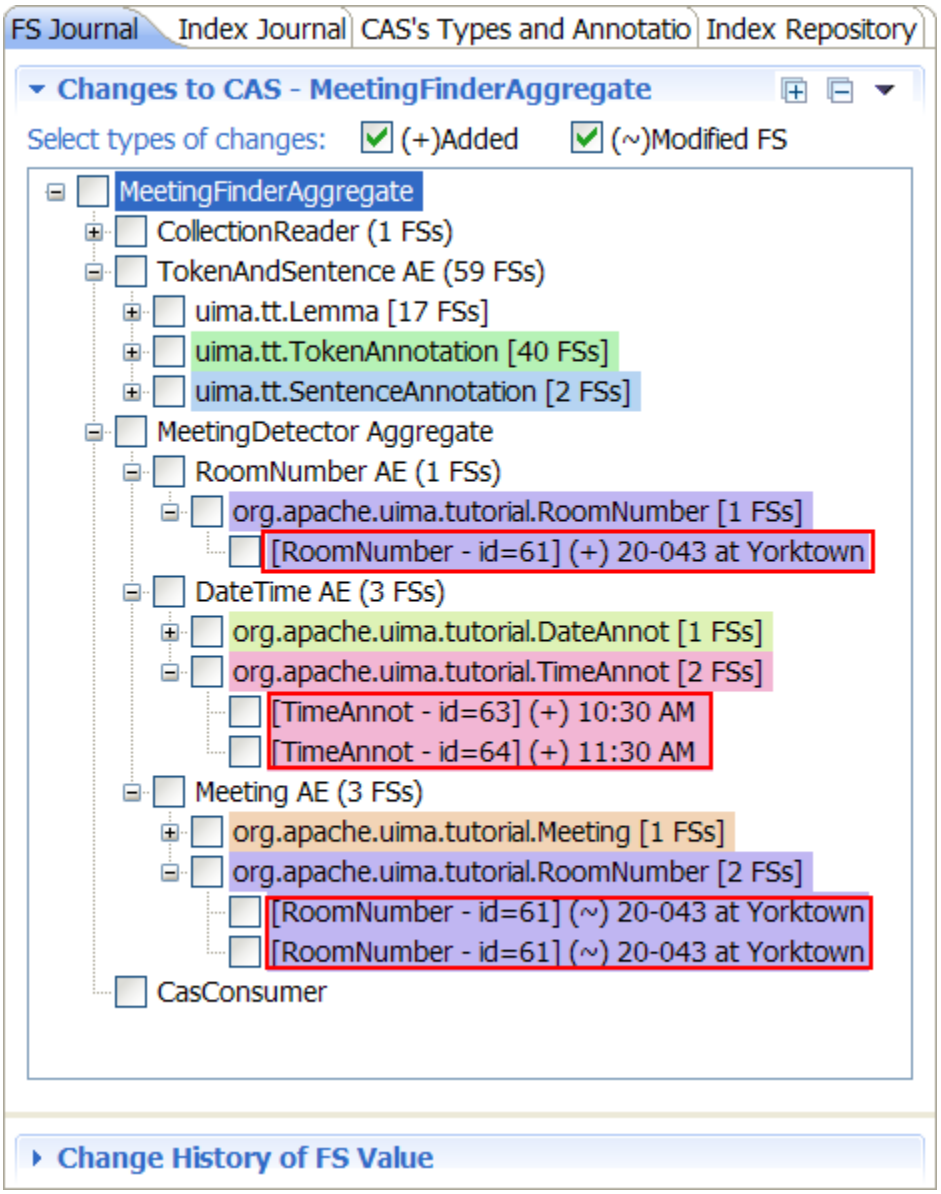

Figure 3.2.b. List of Added/Modified FSs Grouped by Type

Check or uncheck the boxes will control the kinds (added or modified) of FSs to be displayed (see Figure 3.3).

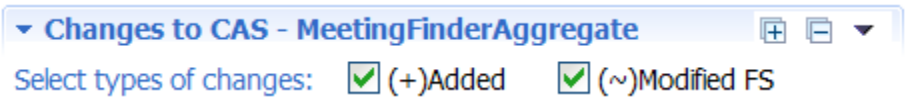

Figure 3.3. Selecting Kinds of FS Changes

Check or uncheck the boxes in the tree will trigger the display of the *annotations*in the input document section. Operations on the input document section will behave the same way as viewing the normal XMI CAS as described the CAS Viewer's user guide.

Note that it is possible to have FSs that are not a sub-type of *Annotation*as shown in Figure 3.3 (the *uima.tt.Lemma*type is defined as a sub-type of *uima.cas.TOP*).

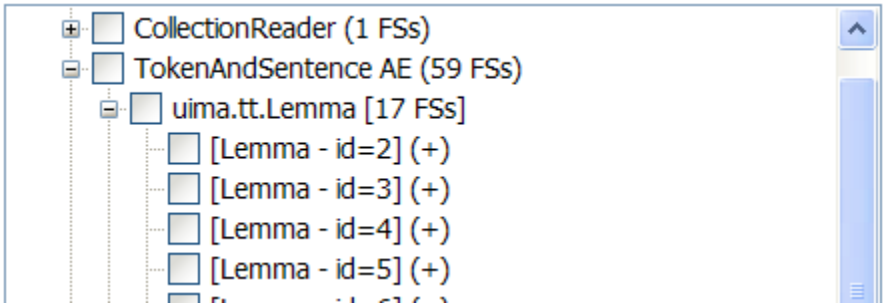

Figure 3.4. Non-Annotation Feature Structures

To view the sequence of changes to a FS, select the FS in the tree. A list of changes (added or modified) by the AEs is displayed in the "Change History of FS value" section as shown in Figure 3.5. The displayed information consists of two parts: the *last* value of the FS and the sequence of changes. The *highli ght* element in the sequence is the FS *selected* in the tree.

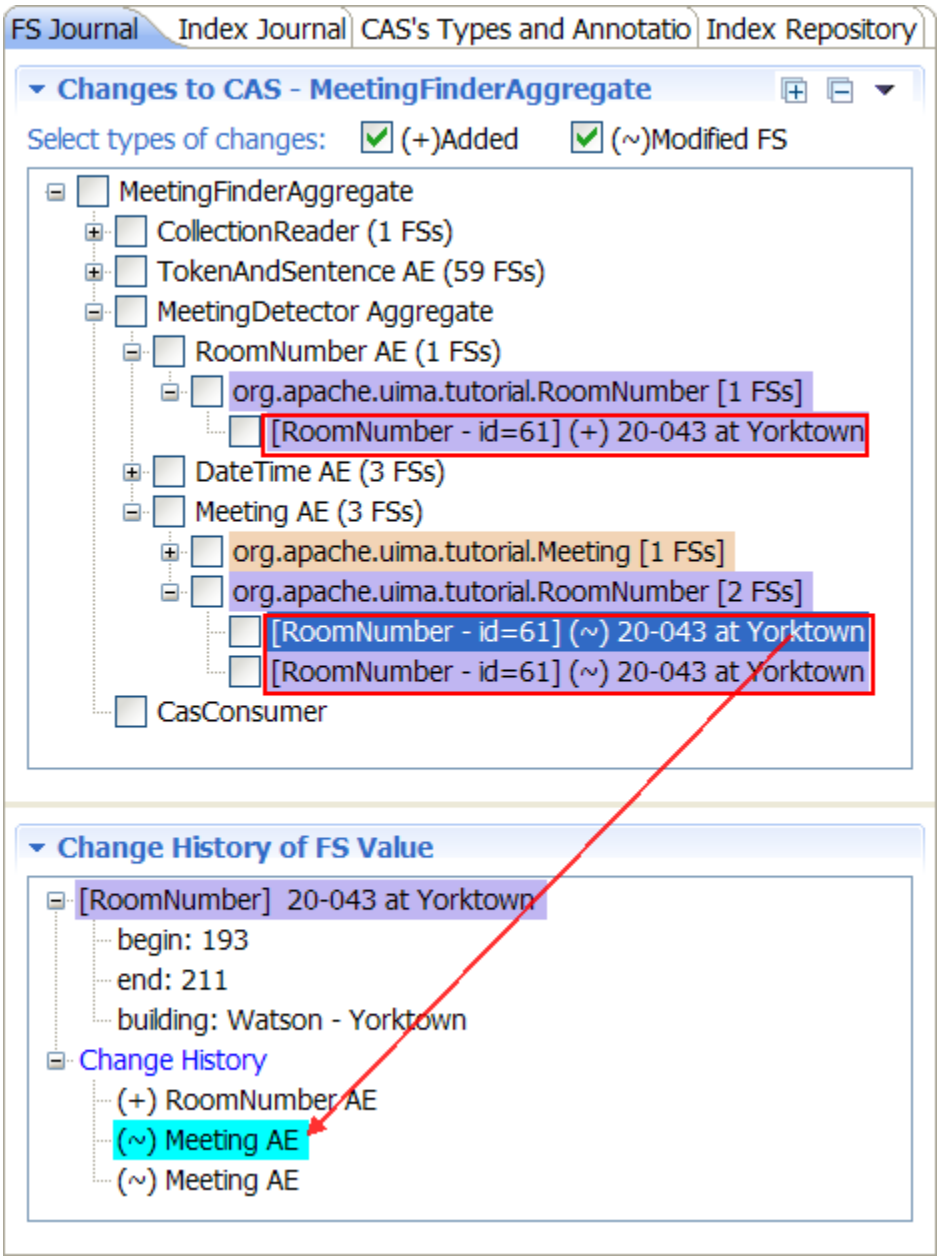

Figure 3.5. Change History of Selected FS - Shows the history for a particular FS, in which component it was created and which subsequent components it was modified.

3.2 Viewing Changes to Index Repository

The information about the changes to index repository is very similar to the changes to the CAS and it is displayed in the Index Journal tab as shown in Figure 3.6. There are two main differences:

- FS Journal tab may contain FSs that are not in the index repository
- Index Journal has an information about the FSs deleted from the index repository and the delete operation followed by an add operation for the same FS will be combined into a modify operation

Otherwise, the operation of the Index Journal tab is identical to the FS Journal tab.

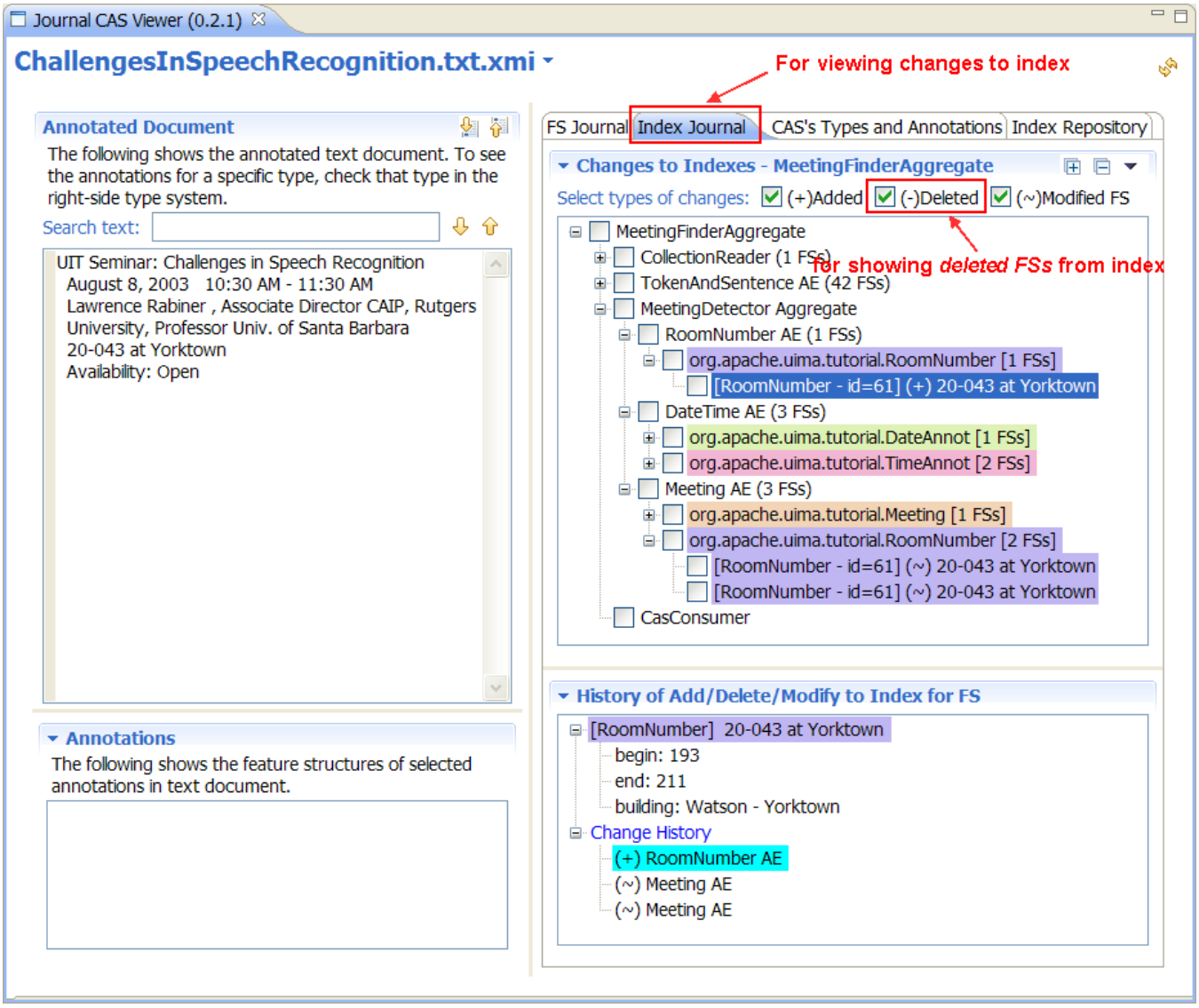

Figure 3.6. For viewing changes to Index Repository