Annex: LLO administration use cases

Source: https://www.surfgroepen.nl/sites/LiLaProject/Shared%20Wiki/Forms/UC%20List%20View.aspx

Use Case create LLO from rig driver

Summary

The LLO creator creates an LLO by using external software (not part of the Portal functionality)

Main actor

LLO creator

Priority

Essential

Preconditions

The LLO creator knows what an LLO is and how to create one. The LLO creator has an already working rig driver.

Main Success Scenario

1. The LLO creator downloads an example LLO (optional).
2. The LLO creator creates locally an HTML page integrating her experiment,
3. The LLO creator places all resources required to execute the experiment locally, in a directory below the HTML content.
4. The LLO creator modifies the imsmanifest file to list all the resources.
5. The LLO creator edits and adds metadata to the imsmanifest file (optional, may be added by the system with UC uploads LLO).
6. The LLO creator packs the data into one .zip file.

Postconditions

The LLO creator has a local LLO ready for uploading (see UC uploads llo).

Use Case upload LLO

Summary

The content provider uploads an LLO from her computer to the system (optionally selecting a rig) and providing metadata describing it.

Main actor

Content provider

Priority

Essential

Preconditions

The content provider has an account on the system; is logged in; has already prepared an LLO (UC creates LLO from rig driver) and has this LLO available on her computer.

Main Success Scenario

1. The content provider accesses the *upload experiment* functionality of the portal.
2. The portal asks the content provider what LLO package (.zip) file to upload.
3. The content provider selects the LLO from the files in her computer.
4. The system validates the LLO, providing feedback to the content provider in case the LLO is ill-formed.
5. The portal provides the content provider with an appropriate interface to add, edit or delete LLO metadata.
6. The content provider uses this interface to enter the required metadata and optionally, additional non-mandatory metadata.
7. The system persists the LLO and its metadata, sets its LLO state as "unpublished LLO" and creates a URL for displaying the LLO content in a web-browser.

Alternative Scenario Extensions

- Extension in step "6": The content provider uploads an RR-LLO that accesses a rig. (There could be RR-LLOs that do not access any rig.)
  1. If the rig is still not registered in the portal, the content provider registers it.
  2. The content provider selects the rig that the RR-LLO uses from the list of her registered rigs
  3. The system adds the rig identifier to the LLO metadata.

(Implementation details: The System creates a Rig with no bookings by default, so initially, experiments cannot be accessed.)

Postconditions

The LLO is available to the content provider for testing (UC test LLO).

The LLO is available to the content provider for publishing (UC publish an LLO). (The system changes the LLO state to “published LLO” to make it accessible for other users.)

Use case edit existing LLO metadata

Summary

The content provider updates an LLO by changing its metadata through the Portal.

The purpose of this is to address errors or typos in the metadata definition, or to add more information; not to upload new LLOs.

Main actor

Content provider

Priority

Low

Preconditions

The LLO has been uploaded (UC uploads LLO).

Main Success Scenario

1. The content provider accesses the portal *edit LLO metadata* functionality.
2. The portal provides the content provider with the list of her LLOs.
3. The content provider selects the LLO.
4. The portal provides the content provider with appropriate interface to add, edit or delete LLO metadata. The content provider uses these interfaces.
5. The content provider confirms the modifications.
6. The system validates the changes.
7. The system persists the changes.

Alternative Scenario Extensions

- Alternative in step "5":
  1. The content provider cancels the modifications. The system does not modify the persisted metadata associated with the LLO.
- Alternative in step "7": the system found validation errors
  1. The system provides feedback to the content provider about the validation errors.
  2. The use case continues in step "4"

Use case test LLO

Summary
The user that has uploaded an LLO makes sure that it works as expected.

Main actor
User

Priority
Low

Preconditions
LLO has been uploaded (UC uploads LLO) but not published (UC publishes an LLO).

Main Success Scenario
1. The content provider selects the *test experiment* functionality.
2. The portal provides the content provider with the list of her unpublished LLOs.
3. The content provider selects the LLO, accesses it and makes sure that it works as expected.

Use case publish an LLO

Summary

The content provider approves an LLO for its public or scheduled use.

While the LLO state is unpublished LLO, the uploaded content is only available for private testing by the content provider and is not visible for the rest of the portal users. This use case ensures that the LLO becomes visible for all portal users.

Main actor

Content provider

Priority

Essential

Preconditions

An LLO has been uploaded (UC uploads LLO) to the portal (with all mandatory metadata fields filled-in).

Preferably, the LLO has been successfully tested (UC tests LLO).

Main Success Scenario

1. The content provider accesses the portal *publish LLO* functionality.
2. The portal provides the content provider with the list of her unpublished LLOs.
3. The content provider selects the LLO that she wants to publish, and the system changes its LLO state to published LLO.

Postconditions

The LLO is visible for all portal users.

After this use case, the LLO is not available for testing anymore.

Note.- *If the published LLO is an RR-LLO that uses an external booking system, the description of how to make it available for users is outside the scope of this document.*

Use case withdraw publication of LLO

Summary

The content provider makes an LLO unavailable for the public.

Note.- Useful for LLOs that access temporarily unavailable rigs (for instance, due to long maintenance or upgrade).

Main actor

Content provider

Priority

Low

Preconditions

The content provider has already uploaded and published the LLO (UC uploads LLO, UC publishes an LLO).

This functionality is only available:

- For LLOs for remote experiments that access rigs “out of order” (not available anymore);
- And for LLOs not published under a “Creative Commons” license.

Main Success Scenario

1. The content provider accesses the portal *withdraw LLO* functionality.
2. The portal provides the content provider with the list of her published LLOs.
3. The content provider selects the LLO to withdraw and the system changes its LLO state to unpublished LLO.

Postconditions

The LLO is only visible to the content provider.

Use case delete LLO

Summary
The content provider removes the LLO.

Main actor
Content provider

Priority
Low

Preconditions
The LLO state of the LLO is unpublished LLO.

This functionality is only available:

- For LLOs for remote experiments that access rigs “out of order” (not available anymore);
- And for LLOs not published under a “Creative Commons” license.

Main Success Scenario
1. The content provider accesses the portal *delete LLO* functionality.
2. The portal provides the content provider with the list of her unpublished LLOs.
3. The content provider selects the LLO to be deleted, and the system deletes all persisted data associated with the LLO.