Chair of
Software Engineering

TrucStudio

Semester project within
Software Engineering Laboratory course
(252-2601-00 P)

By: Patrick Huber
Supervised by: Michela Pedroni
Prof. Bertrand Meyer

Student Number: 01-920-305

ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

inf | Informatik
Computer Science

Information

- ***Semester project within Software Engineering Laboratory course***
- *Project period:* **10/2009 - 01/2009**
- *Student name:* **Patrick Huber**
- *Supervisor name:* **Michela Pedroni**
- *SVN base revision:* **1416**

Project description

edit

Overview and Scope

edit

Lukas Angerer introduced, as part of his master thesis, sharing functionality for the TrucStudio entities and also created a server part. Since then some changes were made - also on the entity model. The most important change was from the tree model to a DAG model, which also gives the possibility to create e.g. a cluster within a cluster. Unfortunately, most of the updates needed in TrucStudio were only made within the core functionality of TrucStudio. The scope of this semester project will be to find the problems introduced by the changes regarding the server functionality, develop possible solutions and implement them. Besides this, there are several open wishes and issues which could be implemented regarding gui / server handling.

Unfortunately, checking and updating the existing sharing features took most of the available time. That reduced the time for adding new features or improvements. The section on future work at the end of this page lists open features or improvements.

Background

edit

Reading list:

- Lukas Angerer: TrucStudio – Truc Sharing; Master Thesis, February 2008
- Florian Geldmacher: TrucStudio – Refactoring clusters; Master Thesis, August 2008

Project Work

edit

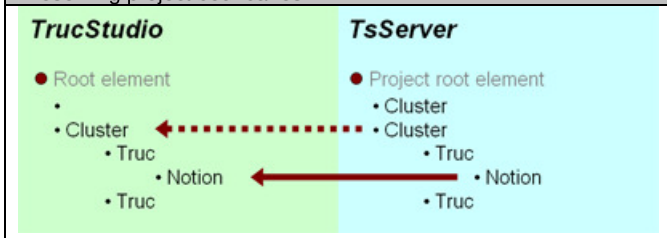
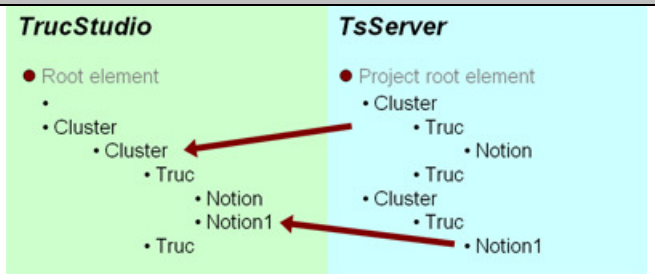
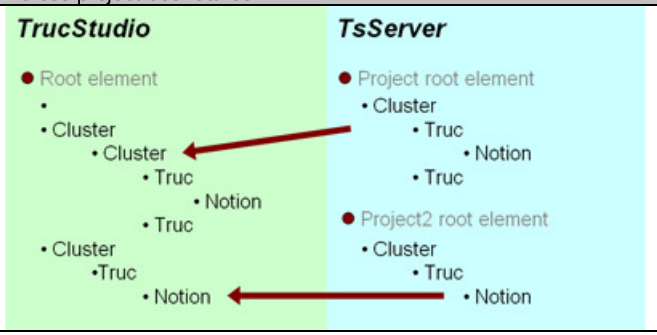
The present work has two main components:
1) Adapting the Truc sharing to the new DAG based model
2) Extending the user interfaces and adding functionality

Adaptation to the new model

edit

The newly introduced model creates some problems in connection with certain server functionalities. Download and also upload was based on the tree based model and happened recursively. The DAG based model rendered these tasks more difficult – so the following summary shows problems and possible solutions.

TrucHandling – Proposition for handling entities in several scenarios	Remark: Dependencies = Links between entities.

Preserving project boundaries		Across project boundaries
		

Characteristic:	Characteristic:	Characteristic:
- Preserve object identity (Notion keeps the same identity) - Preserve graph structure	- Preserve object identity only when parents are equal (Notion1 would be a new created Element) - Download and insert additional elements everywhere	- Preserve object identity only when parents are equal (same as before) - Objects from other projects are newly created.

Updates:		Updates:		Updates:	
(Conservative) - Copy only selected element - Insert stubs for missing parent objects	- Copy whole hierarchy up to the root element (as showed in image)	- Download … … undefined preserving graph structure - insert stubs for missing parent relations (same identity) … defined when allowed to corresponding destination (new idendity)	- Download … … undefined preserving graph structure – copy whole structure for missing parent relations (same identity) … defined when allowed to corresponding destination (new idendity)	- Download … … undefined preserving graph structure - insert stubs for missing parent relations (same identity) … defined when allowed to corresponding destination (new idendity)	- Download … … undefined preserving graph structure – copy whole structure for missing parent relations (same identity) … defined when allowed to corresponding destination (new idendity)

- Uploads / Updates: No problems since we can check if there is an element with the same id.	- Upload to another project?

Effects:		Effects:		Effects:	
- Dependencies are preserved (by definition)		- Dependencies can be preserved (same project)		- Dependencies can be preserved only when in same project	
- Stubs are generated for all parents.	- Children of a root element of a desired entity may also b copied.	- Stubs are generated for all parents (case undefined download).	- Children of a root element of a desired entity may also b copied.	- Stubs are generated for all parents (case undefined download).	- Possibility do keep dependencies within one imported structure.

Problems / Questions:	Problems / Questions:	Problems / Questions
	- Should dependencies copied when an object is created by downloading to an other destination?	- Should dependencies copied? - Upload to another project?

Pros / Cons:	Pros / Cons:	Pros / Cons:
+ Simple for user: Project repository acts as an exact image of the project loaded in trucstudio.	- Copy progress should not be integrated into the upload / download functionality -> Utility of the tool, not of the Server	+ Reusing entities of other projects - Again: Should it be a functionality of the server?

We decided to implement the following solutions:

- Project download
  - Reference copy / download semantic
  - Download directly into root
- Single entity sharing and project based sharing

Restrict the download on a project level has the advantage that the user doesn't have to think about the destination when downloading an entity. Otherwise he should also pay attention to existing parent and child relations (the downloaded is not propagated and therefore these entities not visible) and to new neighbours and references, which would also be updated on the server e.g. when downloading a cluster into a new cluster. I suggest, as mentioned in the future list below, to implement a 'single download' feature with copy semantic. On the other side we decided to enable single entity sharing and project based entity sharing. This gives the possibility to upload entities on a much more granular base instead only upload everything or nothing.

Possible extensions

edit

A project brainstorming in december identified the following additions and improvements to the existing Truc sharing facilities:

- User administration:
  - Remove users
  - Reset password
  - UI: List of users
  - Improve UI
- Project administration
  - Delete project
  - Branch project
  - Move project to another server
  - Rename project
  - UI: List of projects
- Entities
  - Moving from project to project
  - Moving entities from server to server
  - Group handling
    - Branching?

Implementation Details

edit

This part shows the implemented changes in detail – first by describing the new features from a user view – and then by pointing at some relevant (technical/conceptual) changes on the implementation level. Relevant in this context means that the report does not mention or enumerate each changed detail in the code but rather restrict on design and communicational changes. Important classes are mentioned explicitly.

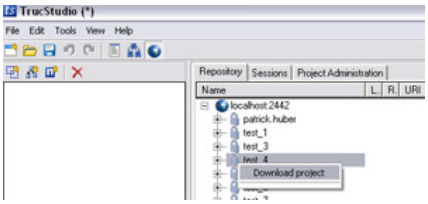
Server functionality: Download project

edit

User guide description

edit

- Added feature: Download project - which downloads all entities (cluster, truc, notion / cluster, lecture) of a project
  - Structure of the graph stays consistent (top elements stay top elements)



Technical description

edit

- Downloading a project happens straight forward: Run through the list of entities and download them when they are not already downloaded
  - Stubs are created automatically where needed (stub: not yet initialized entity)
  - No dummies are created (dummy: temporary entity originating on a server)
  - 'is_top_level' flag is stored on server and used when downloading
- Classes
  - \trunk\src\tstudio\controller\controller_tree\ts_c_cl_tree.e
  - \trunk\src\tstudio\controller\controller_tree\ts_c_truc_notion_tree.e

Server functionality: Share entity

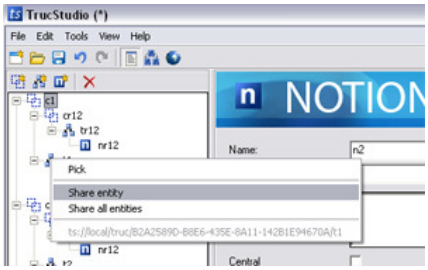
edit

User guide description

edit

- Feature: Share entity

- Single sharing possibility for all entities



Technical description

edit

- Sharing an entity commits in a first step the selected item. In a second step, the non-local entities with references to the newly uploaded entity update this link
  - TrucStudio adds the comment 'Update references' to the commit log of the entities that were updated in step 2 (if any)
- Classes
  - \trunk\src\tstudio\rpc\proxies\ts_entity_repository_proxy.e

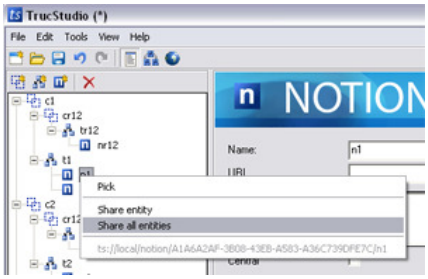
Server functionality: Share all entities

edit

User guide description

edit

- Added feature: Share all entities
  - Sharing of all local entities within the working space
  - Same comment is inserted for all entities



Technical description

edit

- Sharing all entities commits in a first step all collected local items. In a second step, the non-local items with references to the newly uploaded entities update the links
  - TrucStudio adds the comment 'Update references' to the commit log of the entities that were updated in step 2 (if any)
- Classes
  - \trunk\src\tstudio\rpc\proxies\ts_entity_repository_proxy.e

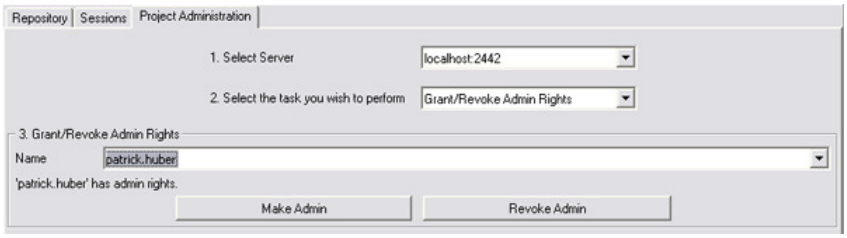
Improved Feature 'Grant/Revoke Admin Rights'

edit

User guide description

edit

- Added feature: A new combobox allows admins to select an existing user



Technical description

edit

- New service 'User management service': 'user.list_all_members'

New RPC-XML request wich only sends the session id as paramter

```
Param # 1
Desc. the session UUID in the format xxxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxx (8-4-4-4-12 hexadecimal digits 0-9 and A-F)
Type string
Example
```

```
    <param>
      <value>
        <string>3F2504E0-4F89-11D3-9A0C-0305E82C3301</string>
      </value>
    </param>
```

The RPC-XML response corresponds to a list of user objects which is returned

- Classes
  - \trunk\src\tstudio\rpc\online_view\ts_project_tab.e
  - \trunk\src\tstudio\rpc\proxies\ts_user_manager_proxy.e
  - \trunk\src\tslib\response_types\ts_user_response.e
  - \trunk\src\tslib\objects\ts_user.e
  - \trunk\src\server\tsserver\data\tss_objects\tss_user.e
  - \trunk\src\server\tsserver\services\user_management_service.e

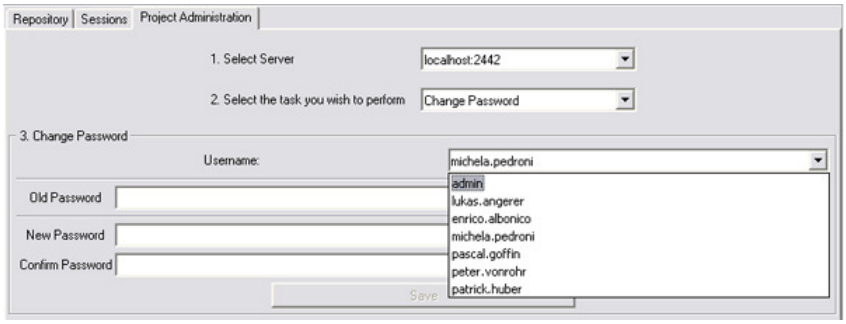
Improved Feature 'Change Password'

edit

User guide description

edit

- Added feature: A new combobox allows admins to reset also the password of other users
- Changing the password of another user is first restricted by TrucStudio and also retested by the server – changing the own password is always allowed



Technical description

edit

- Changes in the 'User management service': 'user.change password'

Within RPC-XML request the new parameter 'username' has to be sent to the server. Corresponding updates are made in the response type and response object classes.

```
Param # 1
Desc. the session UUID in the format xxxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxx (8-4-4-4-12 hexadecimal digits 0-9 and A-F)
Type string
Example
    <param>
      <value>
        <string>3F2504E0-4F89-11D3-9A0C-0305E82C3301</string>
      </value>
    </param>
```

```
Param # 2
Desc. the affected login name
Type string
Example
    <param>
      <value>
        <string>login_name</string>
      </value>
    </param>
```

```
Param # 3
Desc. the old (i.e. current) password
Type string
Example
    <param>
      <value>
        <string>old_secret</string>
      </value>
    </param>
```

```
Param # 4
Desc. the new password
Type string
Example
    <param>
      <value>
        <string>new_secret</string>
      </value>
```

```
    </param>
```

- Classes
  - \trunk\src\tstudio\rpc\online_view\ts_project_tab.e
  - \trunk\src\tstudio\rpc\proxies\ts_user_manager_proxy.e
  - \trunk\src\server\tsserver\services\user_management_service.e

Conclusions and Future Work

edit

The present work adapted the Truc sharing mechanism to the new model that allows a DAG-like structure for entities as opposed to the previous tree structure. In particular, the introduced changes make it possible to download and upload entire projects as well as upload single entities. This enhances the flexibility and ease of use of the Truc sharing facilities. Additional changes improve the administration interface by providing lists of existing users at several points in the administration interface (e.g. for granting/revoking admin rights). A third major improvement now allows administrators to reset the password of other users.

There were no database design changes necessary until now – this in contrast to some statements. The server stores the entities in a local folder and not in the database. The database on the other side maintains the users, projects and also stores entity information like the revision or the log. But some of the features mentioned below might make changes on the database necessary (e.g. introduce an inactive field).

- Design
  - Adapt design changes (correct MVC implementation by F. Geldmacher) to the sharing part of TrucStudio
- User administration:
  - Remove users (maybe introduce a flag to set it to inactive to disallow download)
  - Improve UI
    - E.g. List of users with project rights etc.
- Project administration
  - Delete project (maybe introduce a flag to set it to inactive to disallow download -> old references)
  - Move project to another server (copy/reference semantic)
  - Rename project
  - Branch project
  - Project symbol: should maybe reflect necessary updates of project entities
  - UI: List of projects
- Entities
  - Delete entities (maybe introduce a flag to set it to inactive to disallow download -> old references)
  - Moving from project to project (copy/reference semantic)
  - Moving entities from server to server (copy/reference semantic)
  - Branching