Prof. Lichter

OOSC
WS 2019/2020

Konrad Fögen, Nils Wild

Assignment 6

oosc@swc.rwth-aachen.de

Issued: 06.01.2020 Submission: 20.01.2020 Discussion:23.01.2020

6.1 Frameworks

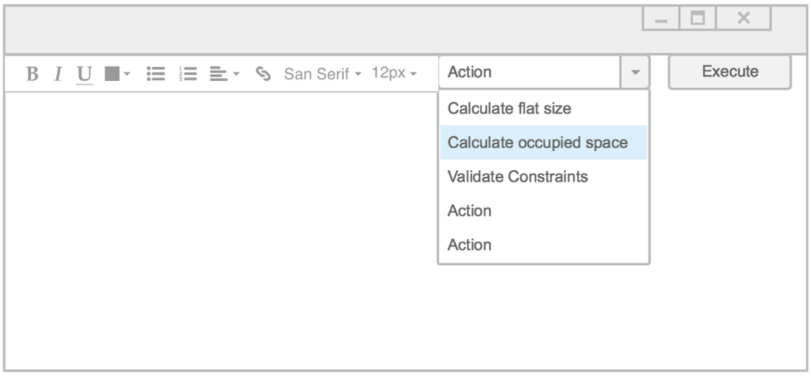
Based on the requirement “User wants to do computations like calculating the size or validating constraints” your team discusses a lot about how to implement these Computations. After a long discussion, you decided to build a framework which supports these calculations but support also other computations/operations like rearrange furniture elements, mass resizing of elements.

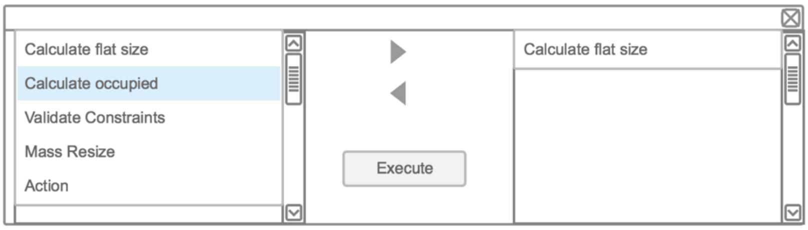
You realized, that you don’t know all possible operations a user might want to execute. A framework seems to solve this problem.

The framework should support the following actions:

- calculate flat size
- occupied space of elements/free space
- validate constraints, e.g. “a room has at least three connected walls”
- Mass operations (resizing of multiple objects)

The Actions should be selectable in dropdown menu and are started with a “Start” Button. In addition Actions can be added to a queue and executed sequentially. Below you find two possible UI dialogs, which visualize this behavior.





Based on your current implementation of the SWCArchitect design and construct a framework to support the Actions described above.

Describe and justify your design decisions and use UML diagrams if appropriate. Describe the properties of your framework and chosen patterns.

6.2. RESTful Web Services

For implementing the “Publish” use-case you are given a software prototype, which consists of a RESTful web service and a web application which uses this web service.

1. Extend the provided prototype to include a API for publishing (adding) Images to the web application. Therefore, extend the current Image Resource. In addition, ensure that the images are persisted correctly and are also available after a restart of the server. If possible use Dependency Injection.
   Please describe and justify your design decisions, components and architecture.
2. Extend your SWCArchitect implementation to act as a client to publish the current drawing to the web service.
   Please describe and justify your design decisions, components and architecture.
3. Implement the Circuit Breaker Pattern to ensure that your client is resilient to a failing web service. **Explain in detail how you implemented the circuit breaker.**
4. Extend the REST-API to approve and deny images. You may change the current prototype by adding new resources, components and domain objects. Please make sure, only approved images are shown in the web application.
   Please describe and justify your design decisions, components and architecture.
5. (Optional) Extend the web application with a new Page, where a moderator can approve or deny images.

The prototype of the web service is provided via a public git repository:

https://git.rwth-aachen.de/swc-public/teaching/swc-oosc-swcarchitect-webservice

The application is able to display images accessible via an URL.

The registered images are not saved and are lost after the restart of the application.

Technical Information:

Build the Application:

- run maven goal package: **mvnw package**
  This will build an executable jar-file located in the target directory.

Execute the application:

- Maven: **mvnw spring-boot:run**
- Execute jar in project home directory:
  java -jar target/webservice-0.0.1-SNAPSHOT.jar

The running application is then accessable via:

- REST: Resource image: **http://localhost:8080/image/info**
- Webapplication: **http://localhost:8080/oosc/app/index.html**