

## 1. Collaborative Reflection of Renovation Process

Beginning with people working in collaboration, a so-called collective intelligence can emerge. Due to interaction and competition, the group has an increased problem-solving capacity. The change of finding a solution within the group is much higher compared to a single person tackling the issues of interest. Especially, the principle of agreeing on reasonable approaches and avoiding critical ideas applies in collective intelligence. The final goal should be consensus decision making. Collective intelligence is measured by the collective IQ. Network effects between distributed data, knowledge, software applications, computing capabilities and experts are used to build up the collective intelligence. Feedback and continuous improvements and learnings are considered in real time. In order to structure the way of working and to put the ideas and comments on record, social media or other contribution systems might be used. For instance, Wikipedia is one platform widely known for collective intelligence, as it allows easy exchange of knowledge, ideas and thoughts.

As already mentioned, collective intelligence is beneficial for solving problems and finding improvements. For this reason, a model wiki based on XWiki<sup>1</sup> allows commenting models and retrieving comments. Furthermore, a kind of review process is included.

### 1.1 Model Wiki application

The Model Wiki web application allow to generate xWiki pages from any model in the ADOxx modelling environment and as soon as the pages are generated allow to import any existing comments in the wiki back to the model.

The user must first have a model available in the ADOxx Modelling environment. As soon as the user created a model or imported in ADOxx, the web application is able to retrieve it and using the “Model Export” user interface allow the user to automatically generate a series of xWiki pages. An external reviewer can then look at the generated wiki pages and collaborate on the model, commenting the relative wiki page. Through the “Import wiki comments” user interface the modeler can decide to automatically import existing comments in a specific attribute of the model or of the model objects.

---

<sup>1</sup> <https://www.xwiki.org/xwiki/bin/view/Main/WebHome>

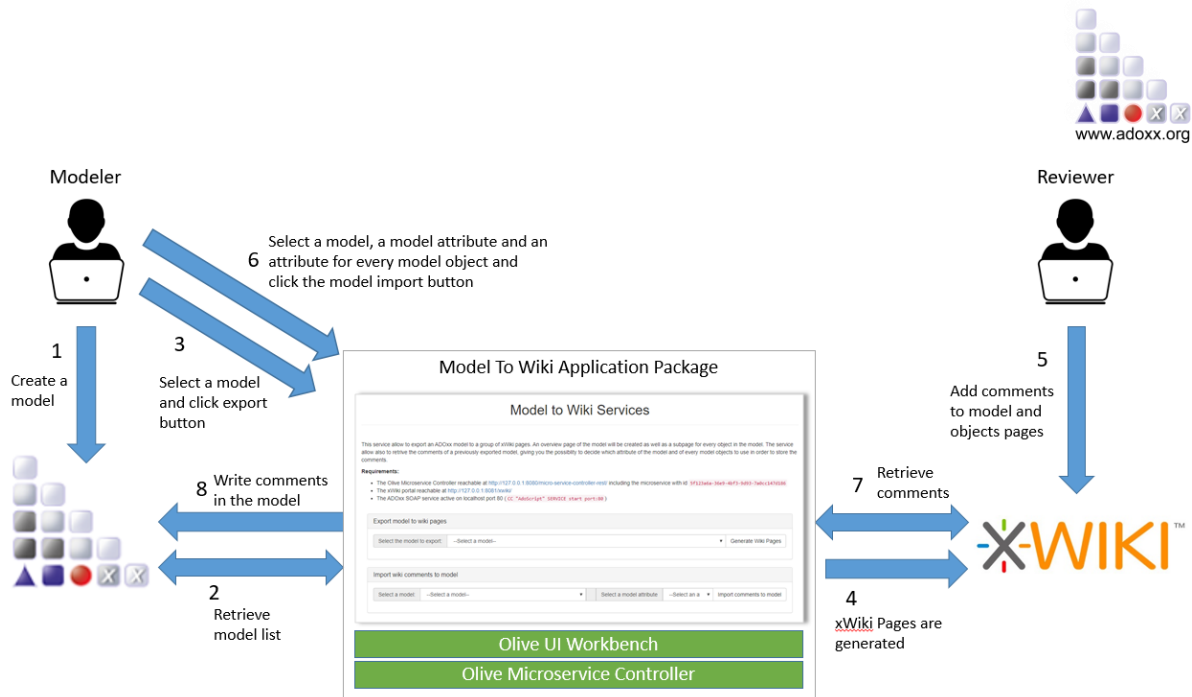


Figure 1 Model Wiki use case scenario

The web application uses the Olive microservices to communicate with the ADOxx modelling environment in order to (1) retrieve the list of all the available models, (2) retrieve all the attributes and objects of a specific model, (3) retrieve the image of a specific model, (4) retrieve all the attribute of a specific object and (5) write the comments in a specific attribute of a model or of an object. Additionally, Olive microservices are used also for the communication with the xWiki platform in order to (1) create an xWiki page and (2) retrieve the comments of a specific xWiki page.

The user interface of the web application has been made using the Olive UI Workbench. In particular the UI is composed of two widgets, named "Model-to-wiki UI" and "Wiki-to-model UI" and displayed in sequence in the main rendering page.

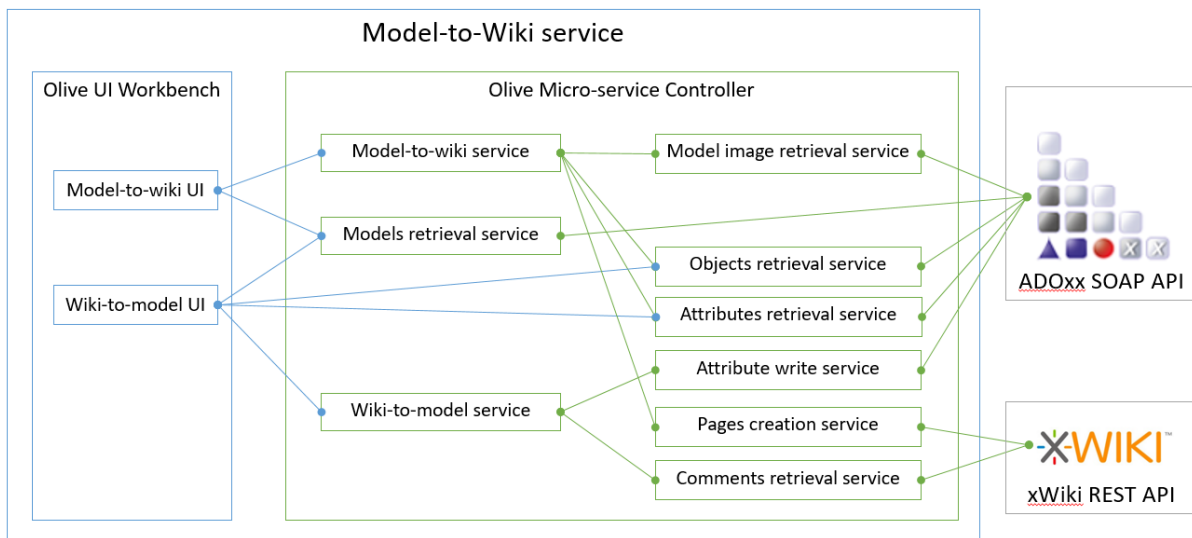


Figure 2 Model Wiki architecture

The xWiki REST APIs are used in order to perform all the required operations in the remote wiki platform.

The ADOxx Modelling environment is instead accessed programmatically using its SOAP APIs that need to be enabled using the AdoScript<sup>2</sup> command `CC "AdoScript" SERVICE start port:80.`

The Model Wiki web application uses the following defined Olive microservices:

- **Model image retrieval service:** microservice that communicate with the SOAP interface of the ADOxx Modelling environment. Require as input a model id and return the Base64 encoded image representation of the model.
- **Models retrieval service:** microservice that communicate with the SOAP interface of the ADOxx Modelling environment and return a list of IDs and names of all available models.
- **Object retrieval service:** microservice that communicate with the SOAP interface of the ADOxx Modelling environment. Require as input a model id and return the list of all the objects inside the model.
- **Attribute retrieval service:** microservice that communicate with the SOAP interface of the ADOxx Modelling environment. Require as input the id of a model or of an object and return the list of all its attributes.
- **Attribute write service:** microservice that communicate with the SOAP interface of the ADOxx Modelling environment. Require as input the id of a model or object, the attribute name to use and the value to write in the attribute. Return a confirmation code that reflect if the attribute has been written correctly or not.
- **Pages creation service:** microservice that communicate with the xWiki REST interface in order to create a page on xWiki. Require as input a page id, page title and page content using the xWiki syntax.
- **Comment retrieval service:** microservice that communicate with the xWiki REST interface in order to retrieve comments on a specific xWiki page. Require as input the page id of the xWiki page to lookup.
- **Model-to-Wiki service:** microservice that orchestrate all the Olive microservices required to generate xWiki pages from a model. The service will generate an xWiki page for the model with information on its graphical representation and its attributes and a sub-page for every model objects containing a description of all the objects attributes. Require as input the model ID and use the “model image retrieval”, “object retrieval”, “attribute retrieval” and “page creation” microservices in the background.
- **Wiki-to-Model service:** microservice that orchestrate all the Olive microservices required to import comments of xWiki pages inside the respective model. Require as input the model ID, a model attribute ID and the list of model object IDs with a model object attribute. Use internally the “comment retrieval services” to find all the comments in the model and use the “attribute write services” to store the comments on the model.

About the frontend side the Model Wiki web application use the following widgets defined using the Olive UI Workbench:

---

<sup>2</sup> <https://www.adoxx.org/live/external-coupling-overview>

- Model-to-wiki UI:** This widget use the “models retrieval service” in order to first obtain the list of all the available models in the ADOxx Modelling Environment and visualize them in a select box, then allow to call the “Model-to-Wiki service” that is responsible to perform all the operations of generating the xWiki pages of the selected model, clicking the “Generate Wiki Pages” button.



Figure 3 Model-to-Wiki UI Widget

- Wiki-to-model UI:** This widget use the “models retrieval service” in order to first obtain the list of all the available models in the ADOxx Modelling Environment and visualize them in a select box. As soon as the user select a model it update the select box relative to the model attributes and the table containing all the model objects with a relative select of object attributes. This done calling the “object retrieval” and the “attribute retrieval” microservices. The model attribute select allow to specify which model attribute to use in order to store comments related to the model (general xWiki page about the model). The object attribute selects allow to specify for every objects which attribute to use in order to store the comments related to the object (specific xWiki sub-page of the model about the object). When the needed attributes have been selected the widget allow to call the “Wiki-to-Model service” that is responsible to perform all the operations of importing the comments for all the xWiki pages related to the selected model in the selected model and objects attributes, clicking the “Import comments to model” button.

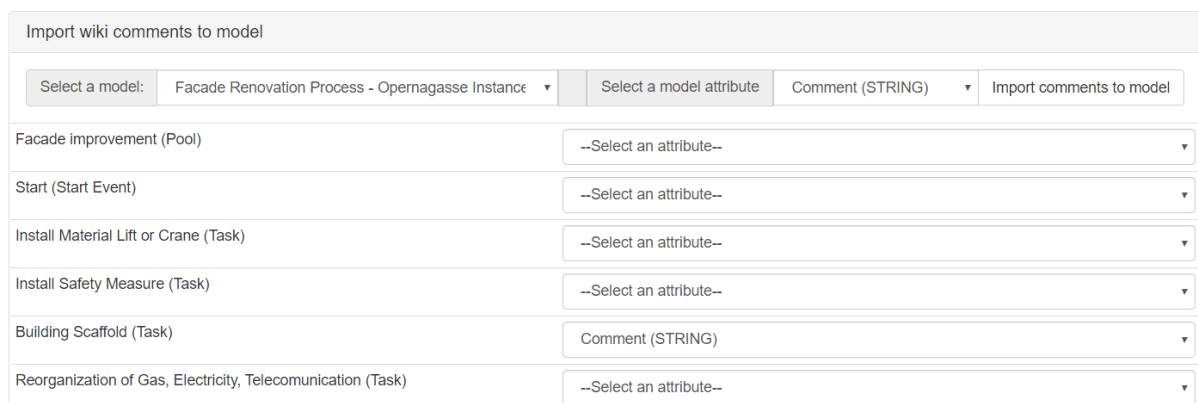


Figure 4 Wiki-to-Model UI Widget

## 1.2 Model Wiki sample use case

This section contain an example of Model Wiki for the Facade Renovation process. This process is modelled in the ADOxx Modelling environment using the BPMN2.0 Library that allow to model processes using the BPMN2 standard notation.

As soon as the SOAP service is started in the ADOxx Modelling environment the Model-to-Wiki UI is able to retrieve the list of all the available models. The user can now select the Facade Renovation process and click the “Generate Wiki Pages” button.

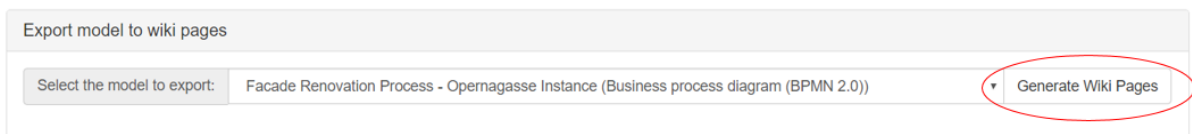


Figure 5 Facade Renovation Process to Wiki

The generation of the xWiki pages may take some times depending on the size of the model. As soon as the generation is completed the xWiki will contain a page describing the process model with its graphical representation and a description of all its attributes. Additionally this page will contain a subpage for every task included in the model with its description.

The reviewer have now the possibility to comment on the page relative to the model or on every sub-page relative to the tasks. In this case a comment have been added on the Building Scaffold task.

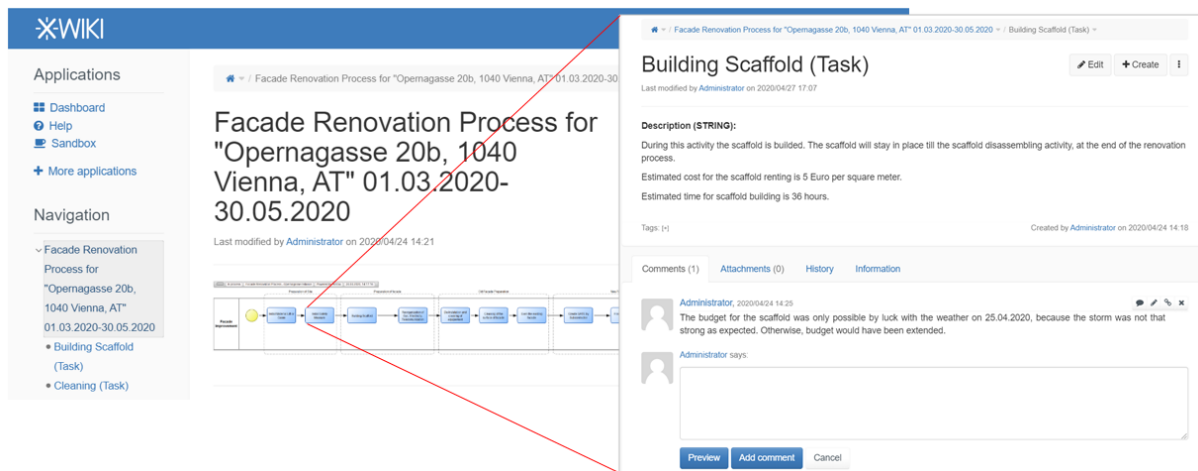


Figure 6 Wiki pages generated for the Facade Renovation Process

When the modeler decide to check the status of the collaboration on the model, he can import the comments into the original model in order to be processed later. The modeler select so the Facade renovation process and the attributes to use to store the comments for every objects (in particular for this case only for the Building Scaffold task).

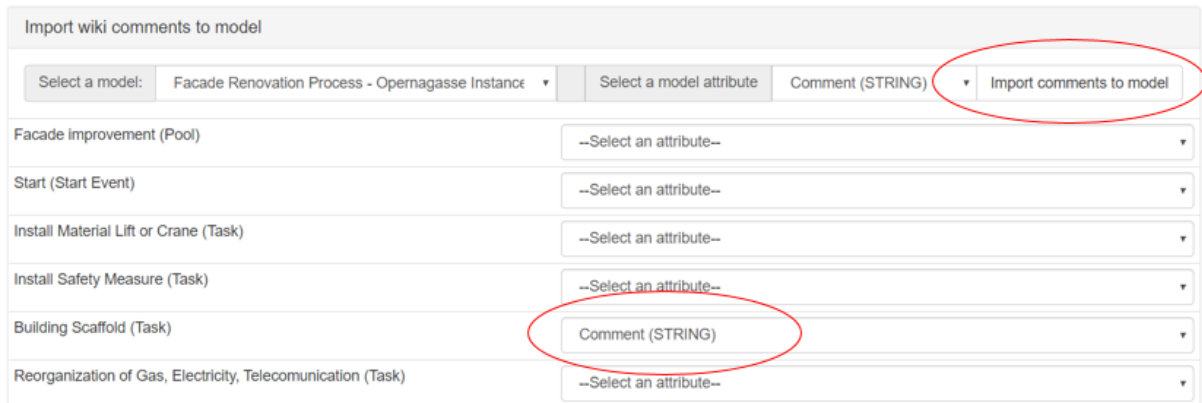


Figure 7 Wiki-to-Model UI widget for the Facade Renovation process

Clicking the “Import comments to model” button the model will be updated and the comments can be visualized in the attribute “Comment” selected. In case of multiple comments all of them will be imported in the same attributes separated by a newline. Information about the user and the timestamp at the creation of the comments are reported as well.

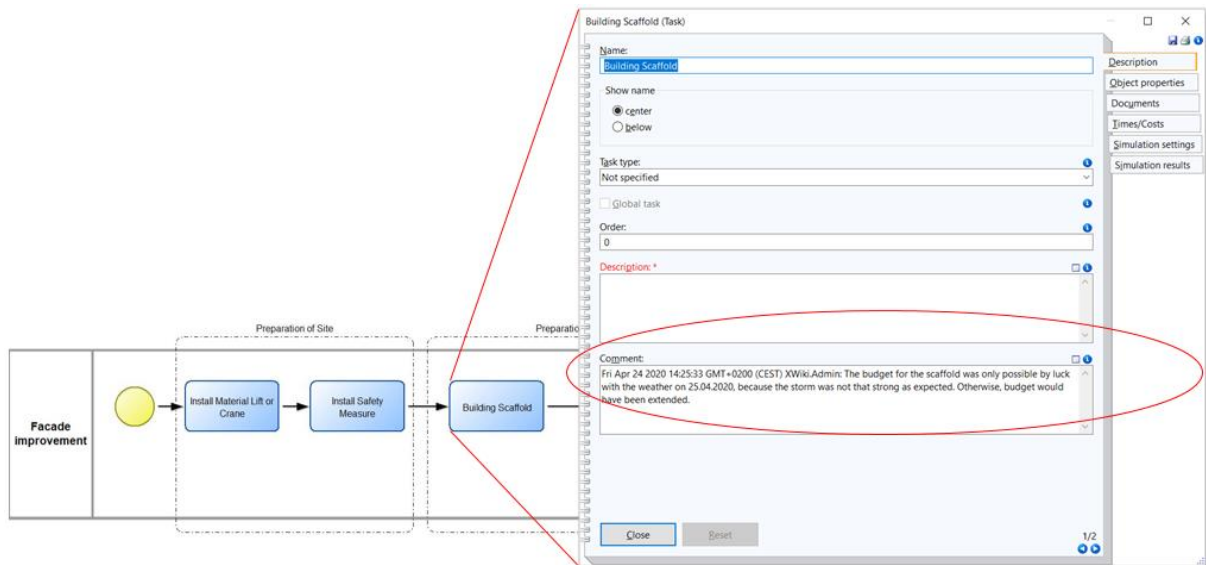


Figure 8 Comments imported for the Building Scaffold task of the Facade Renovation process