

SEALS Semantic Search User-in-the-loop Evaluation Process

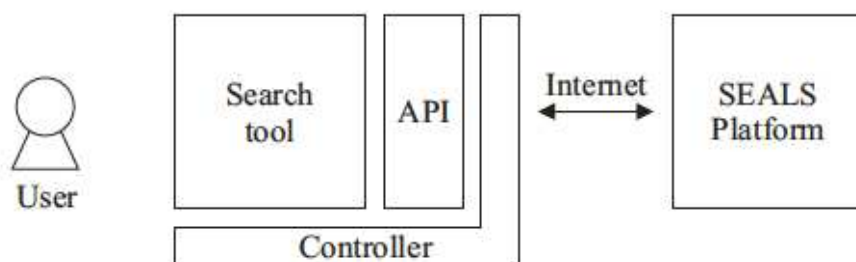


Figure 1: The architecture of the user-in-the-loop controller software.

The user-in-the-loop experiment will be conducted on the tool provider's site where the evaluation will be initiated and controlled by the 'controller' software. As shown in Figure 1, the controller is responsible for any and all communication required with the SEALS Platform such as obtaining test data and also storing results from and to the relevant SEALS repositories. It was also decided that the controller software will be implemented using Java to ensure it is available for as many operating systems as possible.

Instructions for tool providers for using the controller software:

- As part of the tutorial-package folder you downloaded earlier you'll find a folder called "sst-controller".
- Make sure you have already wrapped your tool according to the steps provided in the tutorial "sstool-wrapping-tutorial").
- Rename your tool wrapper jar file into "sealswrapper.jar" and place it together with any other libraries required by your wrapper into the lib folder found inside "sst-controller" folder.
- Run the jar file "sst-controller.jar" using the command "java -jar sst-controller.jar" and you should see the first screen as follows.

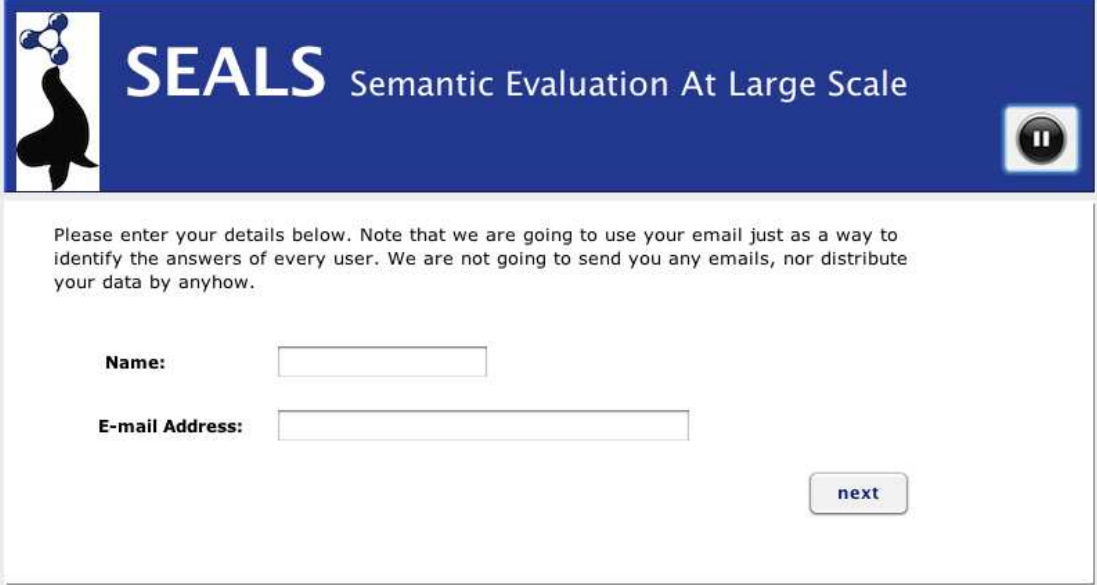
The first screen you get when you run the controller is shown in Figure 2.

The screenshot shows the SEALS Semantic Evaluation At Large Scale interface. The header is blue with the SEALS logo and text. Below the header, there is a section titled "This is for the Experimentation Leader". It contains two prompts: "Please select the deployment path for your tool wrapper" and "Please enter the class name of your tool wrapper". The first prompt has an "Open" button and a text input field. The second prompt has a text input field. A "Finish" button is located at the bottom right.

Figure 2: Tool Wrapper Form

This is the only screen requiring input from you as a tool provider rather than from the user. For the first input field, you need to either enter or browse and choose the path where your tool wrapper is deployed on the machine. The controller does not use this path, rather it is passed to the tool wrapper in case it needs to know its deployment path. If you know your wrapper will not use it, you can just enter "." in this field. The second input field asks for the name of the class that is implementing the functionalities of your tool wrapper (e.g.: *uk.ac.shef.dcs.oak.search.ksearchapi.wrapper.KSearchWrapper*). After the first time you use the controller and enter these values, the controller will remember them and provide you with auto-completion to ease this process.

Next, the user is expected to continue from here, first presented by a welcome screen providing brief information regarding the SEALS project and community. Then as shown in figure 3, the next form asks the user to enter his name and email address that are used for the purpose of identifying users and mapping them to their experiment results.



The screenshot shows a web form titled "SEALS Semantic Evaluation At Large Scale". The header is dark blue with a logo on the left and a pause button on the right. Below the header, there is a white box containing the following text: "Please enter your details below. Note that we are going to use your email just as a way to identify the answers of every user. We are not going to send you any emails, nor distribute your data by anyhow." Below this text are two input fields: "Name:" followed by a text box, and "E-mail Address:" followed by a text box. At the bottom right of the form is a button labeled "next".

Figure 3

Following is a form containing brief instructions for the user for completing the experiment such as "You have the right to suspend or exit the experiment at any time". More detailed instructions regarding the experiment and the tested tool would be provided in some hard copied format. The user is now ready to take the experiment and questions are shown one after the other like in Figure 4.

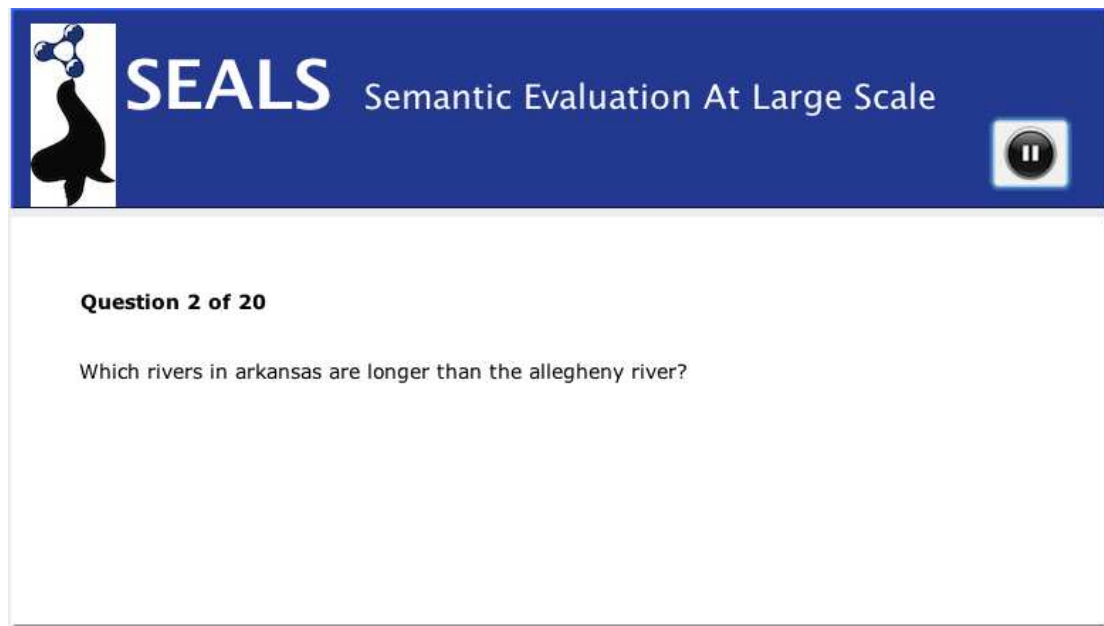


Figure 4

Then, at this time the user would switch to the tool (that is expected to be running) to formulate his query for this question using the tool's interface. When the tool shows the results for the user, the controller captures this event and asks the user if he successfully found an answer for his query or not as shown in the dialogue in Figure 5.

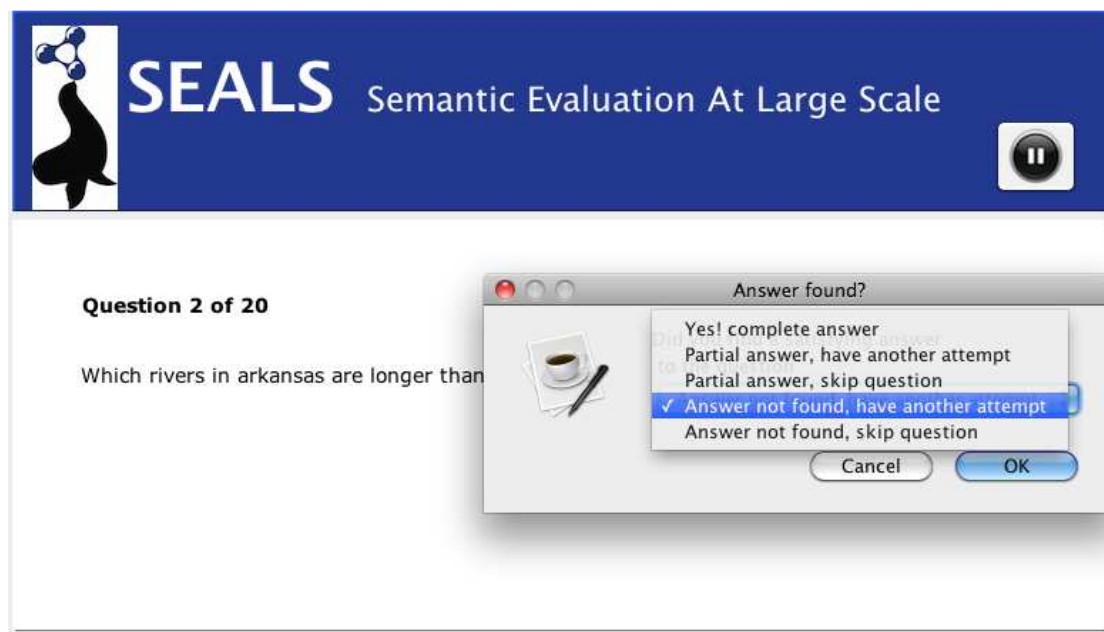


Figure 5

The feedback from the first evaluation campaign showed that users asked for the ability to choose partial satisfaction with the results rather than only being able to choose “Yes” or “No”. Therefore, the new version of the controller provides the user with 5 different choices. The user can be completely satisfied with the results and choose “Yes! Complete answer” or chooses to have another attempt to get more satisfying results or skip the question wither with partially satisfaction or finding no answers for his question.

The pause button that shows in all forms gives the ability for the user or the test leader to pause the experiment and in-turn the timings for a certain amount of time as shown in figure 6. Clicking “OK” will resume the experiment.

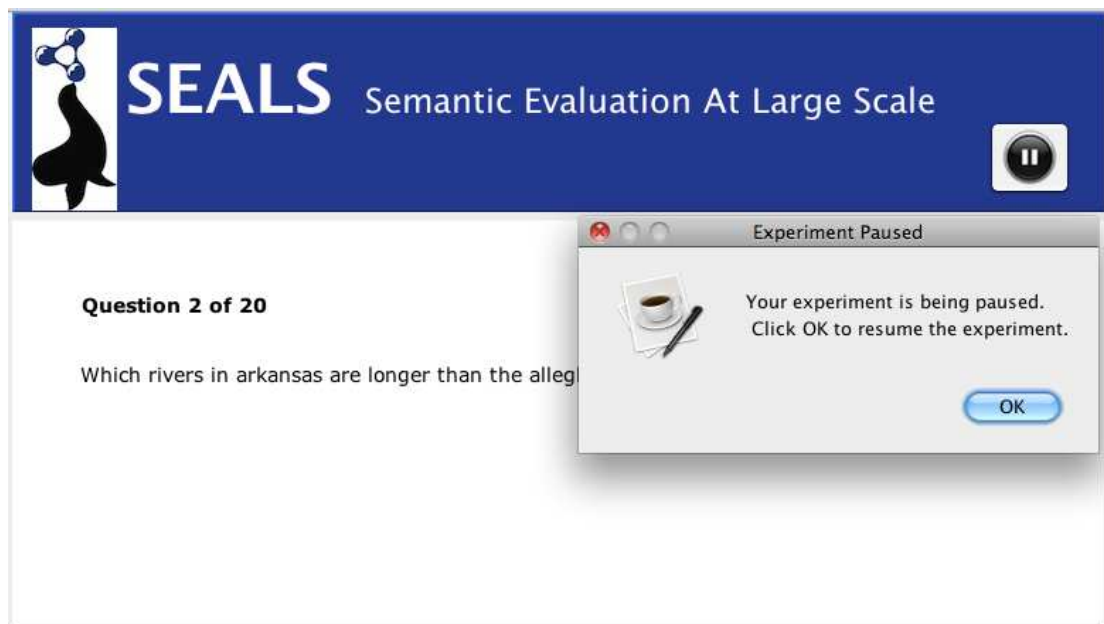


Figure 6

This process continues till the user finishes the list of questions, when he is then presented with the forms that point him to the three questionnaires (SUS, extended and demographic questionnaires) one after the other. By filling the questionnaires, the experiment will get to the end where the controller shows the user the final form, which is a thank you form.

Tips:

If at any point, the user chose by mistake to end the experiment or the tool crashed, the controller provides the ability to continue the experiment from the last stored form. Start the controller again in the same way as explained above and your user will be able to continue his experiment with no loss of results.