

QuASE Tool User Guide

Version 1.3

Klagenfurt 2016

Contents

- Introduction..... 3
- 1 Logging in..... 3
- 2 Startup screen 4
 - 2.1 Synchronizing the knowledge base 5
 - 2.2 Other knowledge base operations 5
- 3 Understandability Management 5
 - 3.1 Selecting context 6
 - 3.1.1 Home context 8
 - 3.2 Selecting the document..... 9
 - 3.3 Translating the document 9
 - 3.4 Specifying the knowledge domain 11
 - 3.5 Translating arbitrary text..... 12
 - 3.6 Understandability assessment 12
- 4 Analysis..... 14
 - 4.1 Collecting external information..... 14
 - 4.2 Aggregate metrics 19
 - 4.3 Similarity search 20
 - 4.4 Predicting metric values 22
 - 4.5 Collecting decisions and recommendations..... 23
 - 4.6 Predicting decisions and issuing recommendations 24
- 5 Additional functions 25
 - 5.1 Bookmark management 25
 - 5.2 User management 26
 - 5.2.1 Changing account information for non-administrator users 28
 - 5.2.2 The QuASE role system..... 28
 - 5.3 Internal endpoint..... 29
- 6 Integrating QuASE and Jira 30
 - 6.1 Working with issues..... 30
 - 6.2 Working with context elements 32

Introduction

In this section of the Guide we describe the standalone version of QuASE, the embedded version is described later in the separate section, most of the information provided here is also applicable to the embedded version; the differences will be outlined later.

1 Logging in

QuASE tool is a Web application accessible through a URL. To start the session, it is necessary to go to the QuASE tool URL in a browser. Compatible browsers are Mozilla Firefox, Chrome, and others supporting up-to-date Web technologies.

On startup, the user sees an information screen which contains the short description of the system and the list of developers. The login link is in the top right color of the screen.

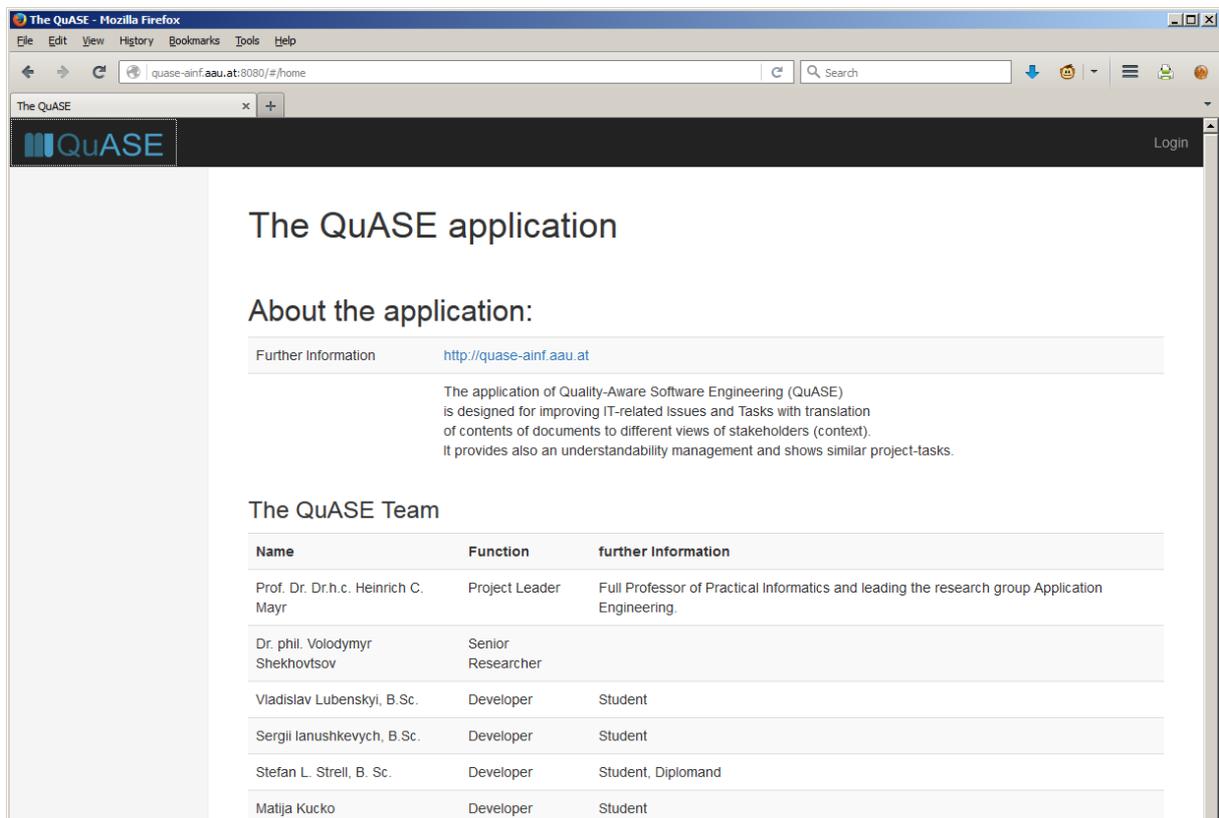


Fig. 1. QuASE information screen

After following the login link the user has to specify the user name and the password:

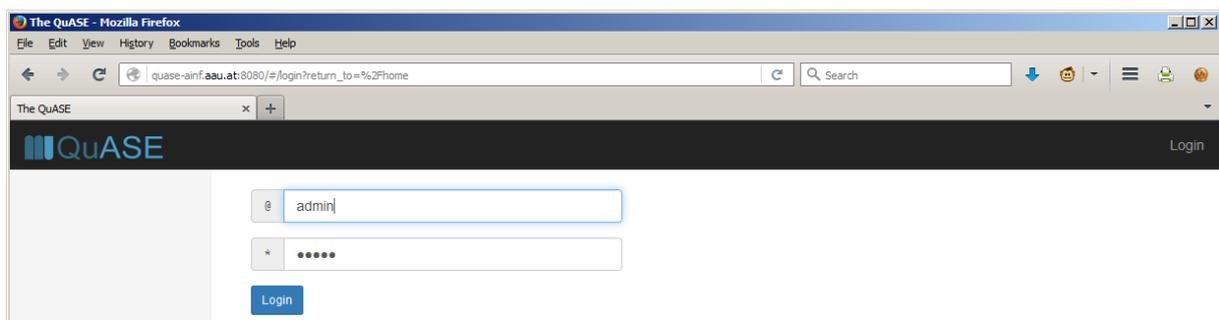


Fig. 2. QuASE login screen

2 Startup screen

After successful login, the user will see the initial startup screen of the tool:

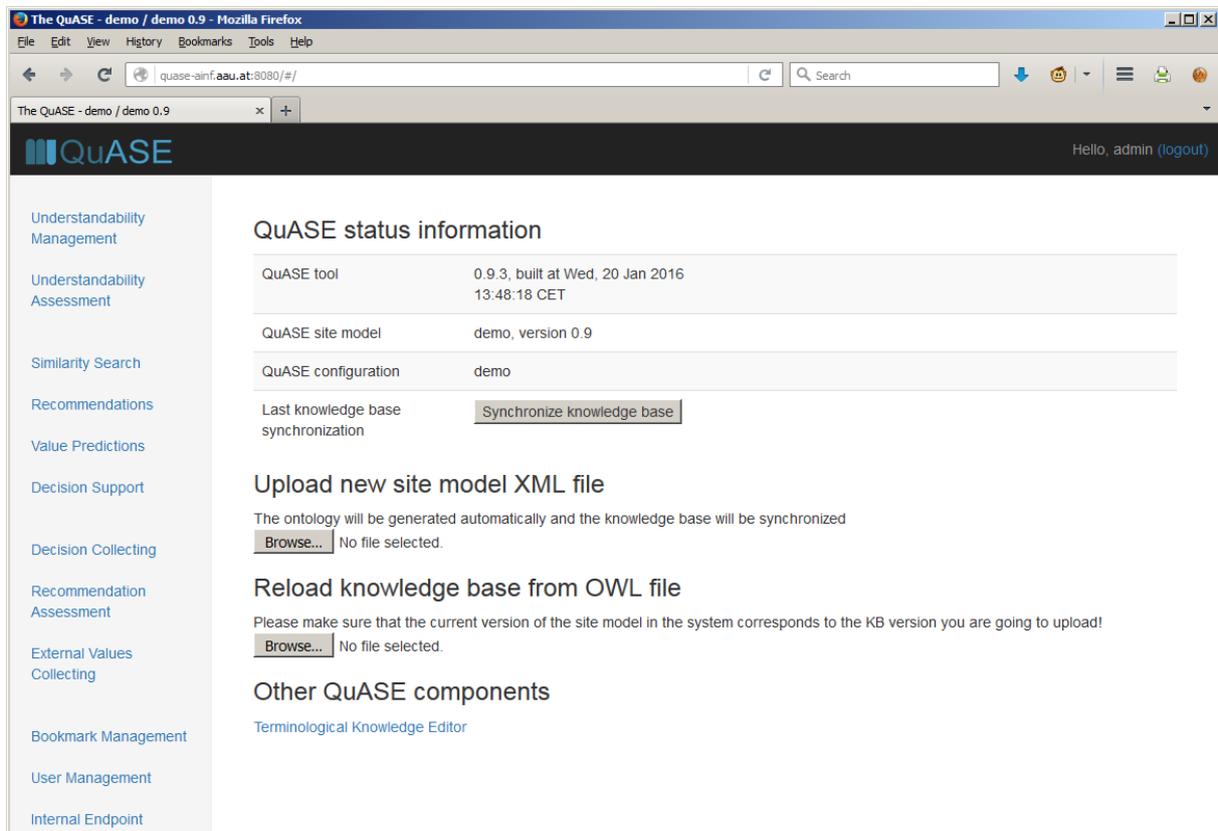


Fig. 3. QuASE initial startup screen

Main QuASE screen is split into three areas:

1. information area on the top of the screen (with black background), it contains QuASE logo and the login link, clicking on the QuASE logo at any time navigates back to the initial startup screen;
2. main menu at the left of the screen, following the links in this menu activates different functional tasks (these tasks are described in detail in the subsequent sections);
3. content area to the right of the main menu, the information displayed there is different for different tasks

For the initial startup screen the content area displays the following information

1. The version of the tool and the time of its last build
2. The name and version of the current site model used to define the structure of the knowledge base
3. The name of the current site configuration
4. The time of the last synchronization of the knowledge base

In addition to this information, the screen also shows links to some actions involving the QuASE system as a whole, the explanation of these actions is provided below.

On the bottom, the initial startup screen also contains the link to the QuASE Terminology Editor Tool, by following this link, the tool will be launched in a separate browser tab.

2.1 Synchronizing the knowledge base

The synchronization of the knowledge base is a process where the information in the project repository (e.g. Jira database) which has been modified since the previous synchronization is transferred into the QuASE knowledge base. By default, the synchronization can be scheduled by specifying the environment for QuASE tool, it can be also launched manually by clicking the “*Synchronize knowledge base*” button on the initial startup screen.

During synchronization (or other actions leading to the modification of the knowledge base) the system displays the “under maintenance” message:

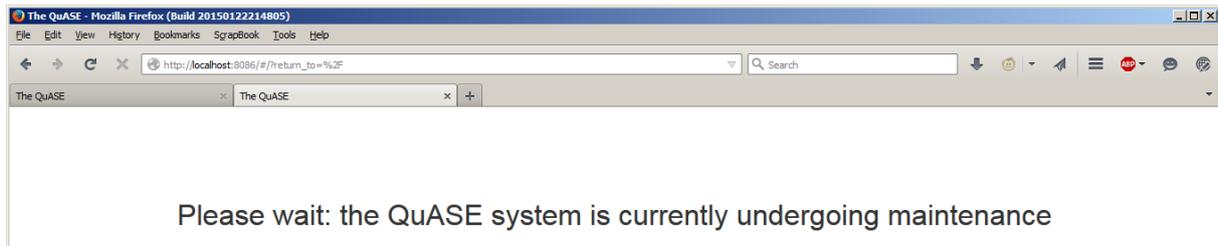


Fig. 4. QuASE “under maintenance” message

2.2 Other knowledge base operations

The initial startup screen also contains the following links

1. “*Upload new site model XML file*”: following this action makes possible uploading new XML file produced by QuASE site modeler. After that the knowledge base will be completely regenerated based on the current version of the project repository and internal QuASE database. The effect of this action is the same as initiating regeneration of the knowledge base from the QuASE site modeler.
2. “*Reload knowledge base from the OWL file*”: following this action makes possible uploading new OWL file containing the complete knowledge base, after this, the full content of the knowledge base will be replaced with the contents of this file. This action is reserved for the cases when generating the knowledge base in the current configuration is not possible for some reason.

3 Understandability Management

After following the “Understandability Management” link in the main menu, the user sees the initial context and document selection screen:

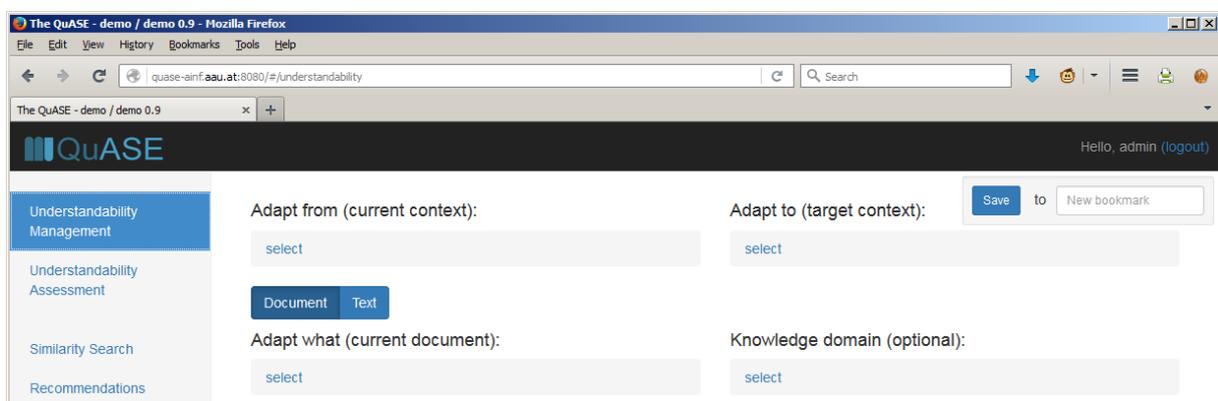


Fig. 5. Initial context and document selection screen

On this screen, four selectors are available (identifiable by the clickable word “select”), these selectors are used for specifying source and target context, document for adaptation, and knowledge

domain. Here, we start from describing the context selection (it is not really necessary to start from context selection while working with QuASE, it is possible to start from document selection as well).

3.1 Selecting context

The context selection screen can contain several control elements for selection. Every such element contains type selector and individual selector. The type selector is a dropdown box on the top of the element (Fig. 6): it allows selecting the type of the context unit (“Person category”, “Project” etc.)

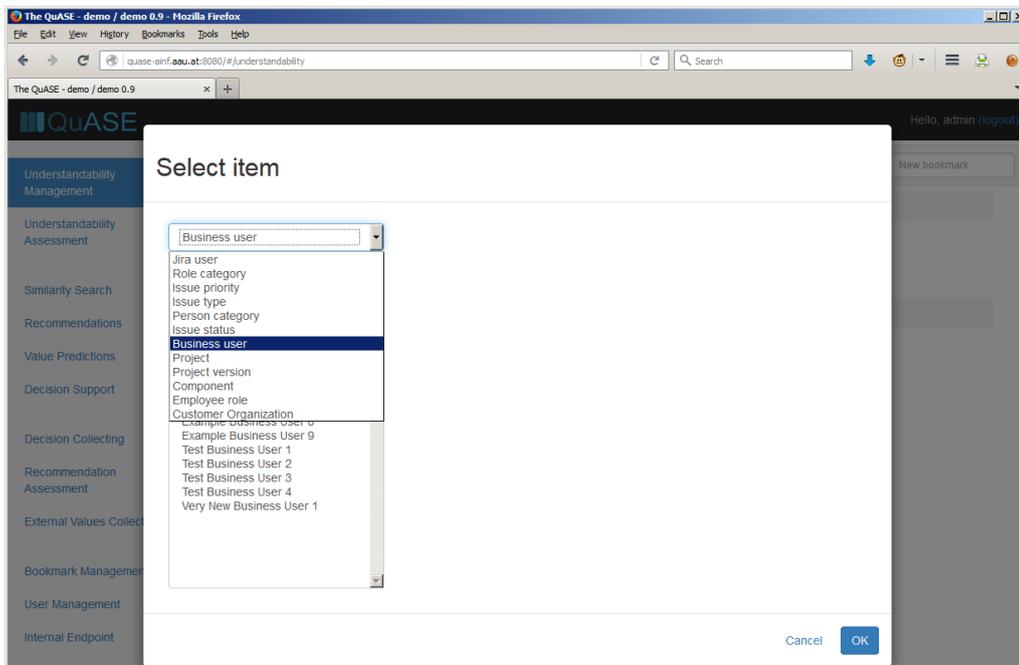


Fig. 6. Context selection: selecting the type of the context unit

After selecting the particular type, the next step is to select the individual of this type by means of the individual selector (Fig. 7). Such selector is a list box displaying the list of individuals of the selected type.

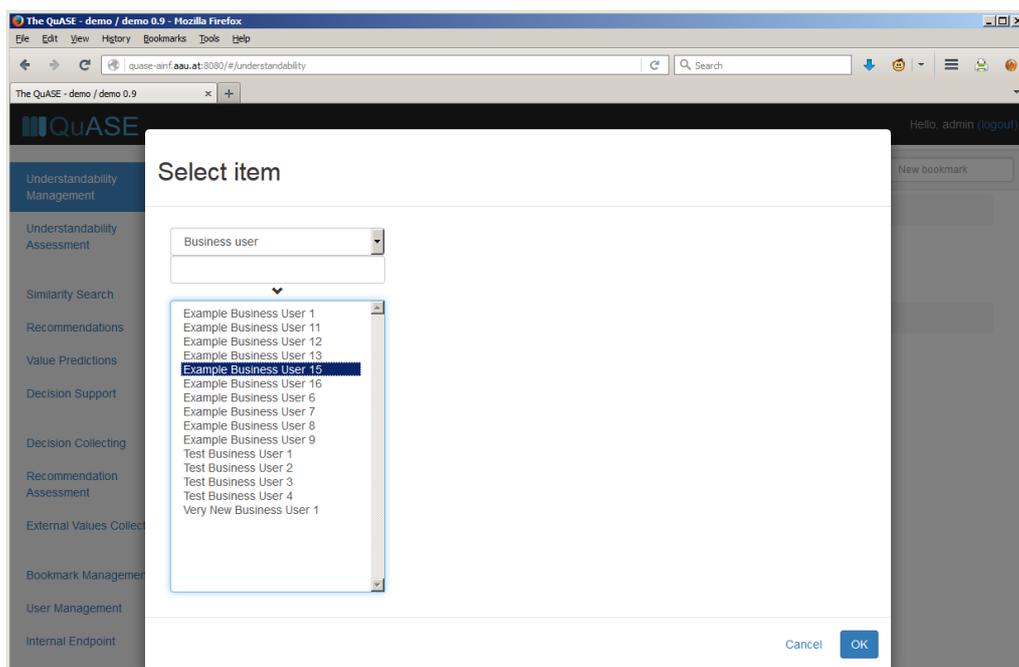


Fig. 7. Context selection: selecting context individual - first level

If the individual selected at the particular level has connections to other individuals (such as the connection between the project and the involved users etc.), the selection screen shows additional control element which again allows selecting the type of the element and the particular individual of this type. Fig. 8. shows two additional levels of selection, here it is possible to select the employee roles belonging to the particular role category, and the particular user belonging to the selected role.

With multi-level selection, it is possible to select the individual on the level higher than the last displayed, and make this selection final by clicking on OK, in particular, on Fig. 8, the selection only involves particular employee role and not particular user, clicking on OK in this case will lead to the particular role being selected.

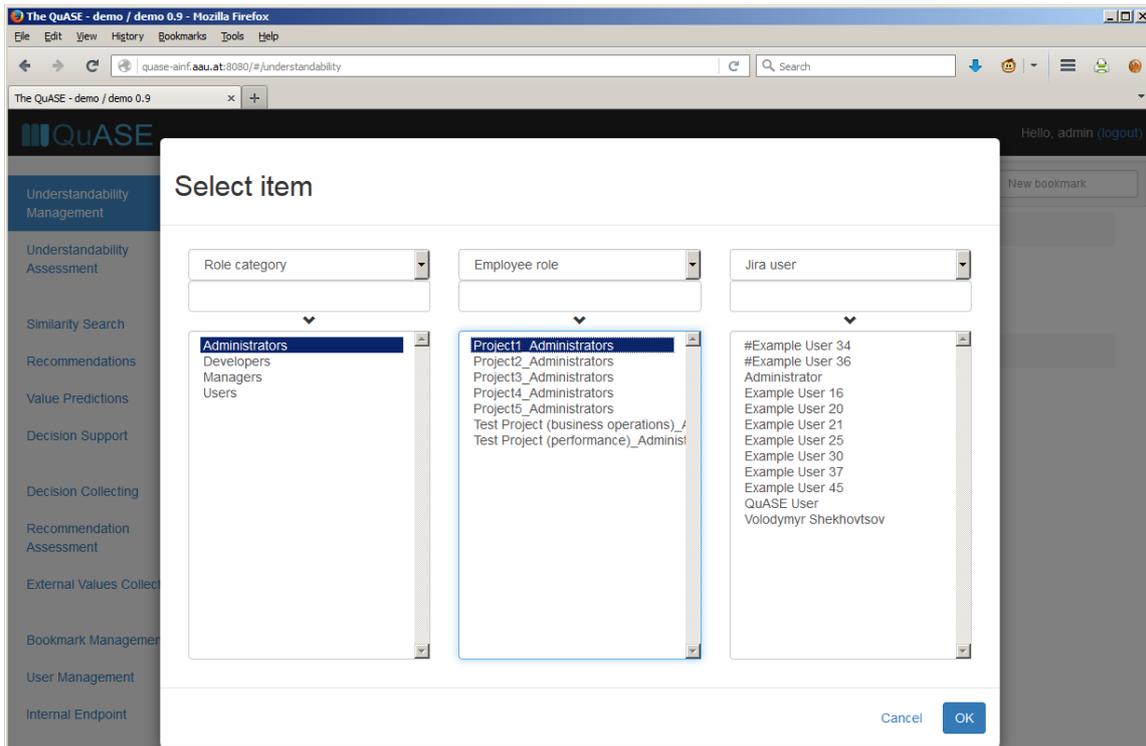


Fig. 8. Context selection: selecting context individual - further levels

After completing the selection, the selection screen will show the selection results (Fig. 9). Clicking on the additional buttons to the right of the name of the selected element will allow editing the selection or deleting it, in the latter case the selector will be brought to the initial state. Clicking on the name of the selected element allows editing the selection, clicking on the cross sign to the right of the name clears the selection, in this case the selector will be brought to the initial state.

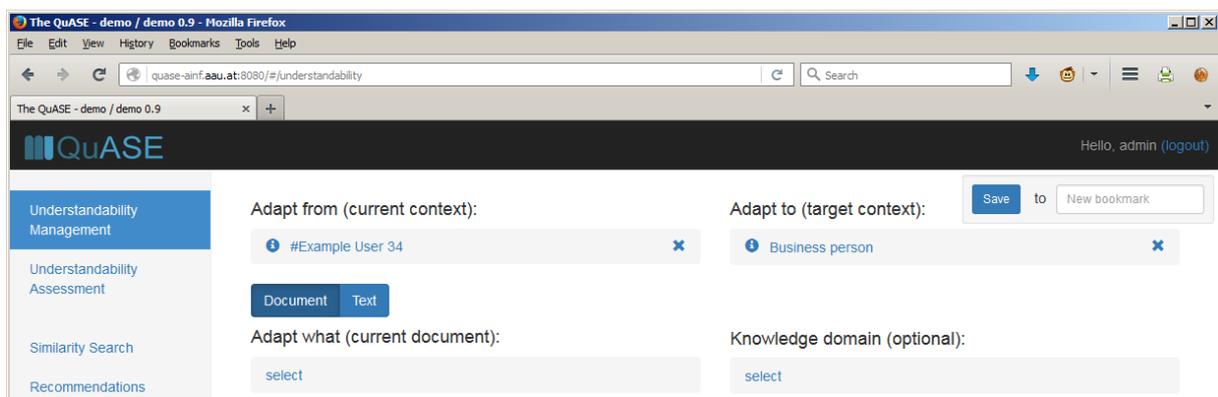


Fig. 9. Context and document selection screen after selecting source and target contexts

Clicking on the “i” sign to the left of the name opens the window with the information about the selected element which includes the name of the element and the values of its attributes (Fig. 10).

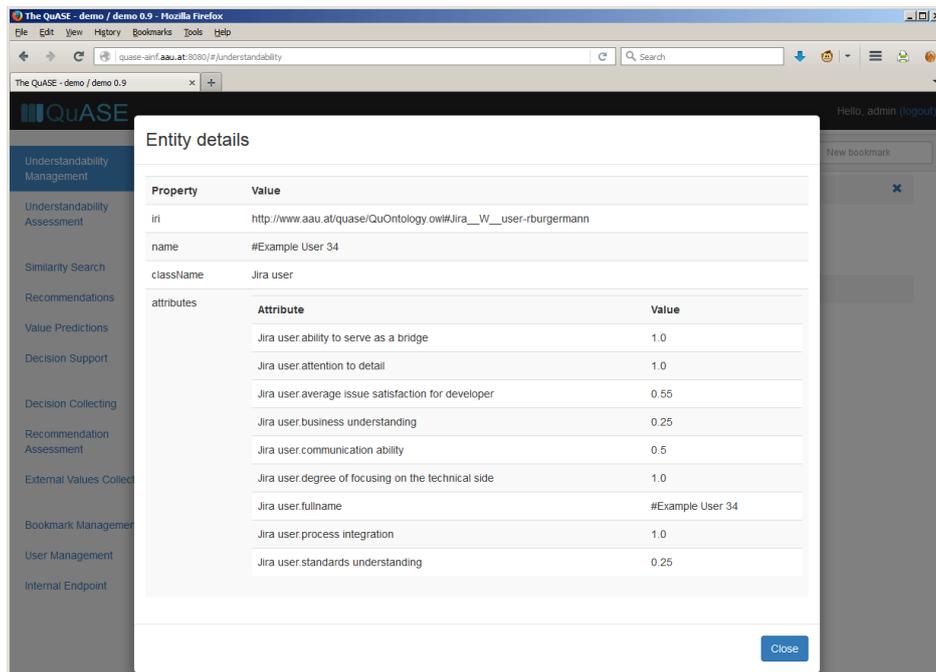


Fig. 10. Information about the selected element

3.1.1 Home context

If the QuASE user is connected to the particular Jira user (this connection is explained in more detail later), after logging into the system, such user can use context selection shortcut defined as *home context*. This context is the Jira user corresponding to the logged-in user, selecting this context allows for the users to obtain immediate access to their own documents.

For such connected user, the context selection screen looks as shown on Fig. 11: the additional “home” sign is visible to the left of the selector.

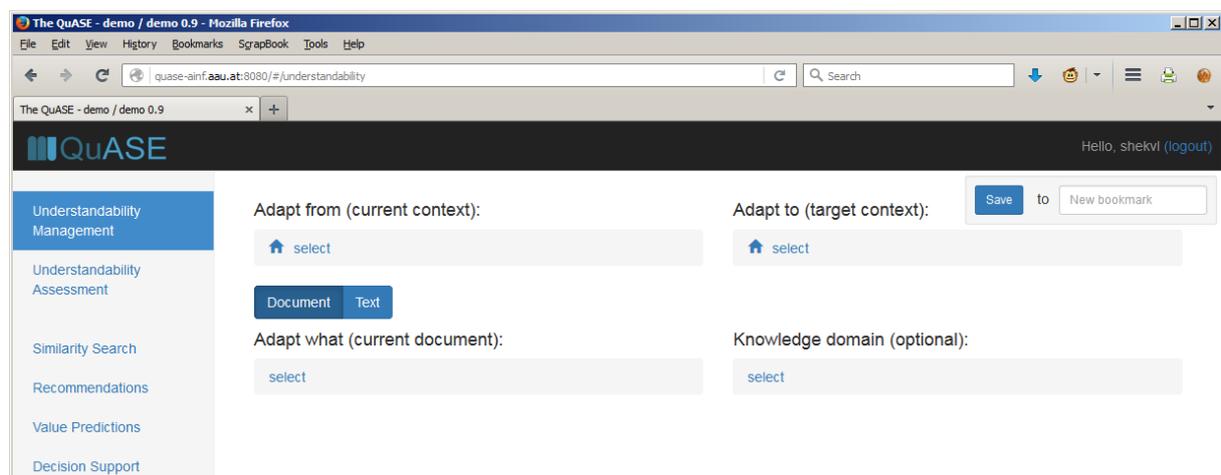


Fig. 11. Context selection for QuASE user having the home context

Clicking on this sign immediately selects the context connected to the current user as shown on Fig. 12, where the selected context (“Volodymyr Shekhovtsov” as a Jira user) is connected to the current QuASE user *shekvl*.

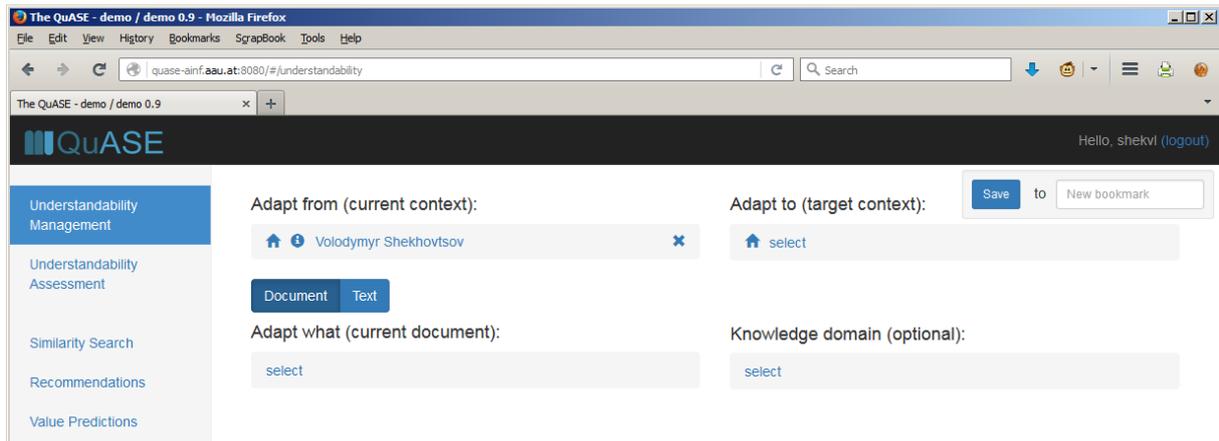


Fig. 12. The home context after selection

3.2 Selecting the document

Selecting the document is similar to selecting context (Fig. 13). If prior to selecting the document, particular source context has been already selected, the displayed documents will be filtered based on this context (only the documents connected to this context will be shown), if no source context has been selected, all documents will be available for selection.

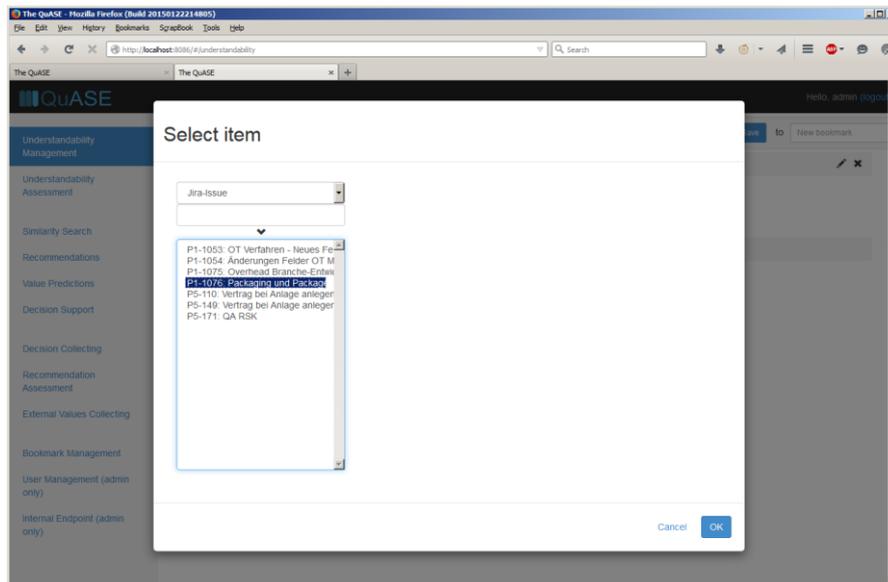


Fig. 13. Document selection

3.3 Translating the document

The prerequisites for translating the document are the selections of the source and target contexts and the document to be adapted, these selections can be performed in arbitrary order; the translation is performed instantly after changing any of these selections. For example, after selecting source and target context, selecting the document immediately leads to displaying its translation, later, if e.g. the new target context is selected, the new translation is shown immediately again. The translation screen is shown on Fig. 14.

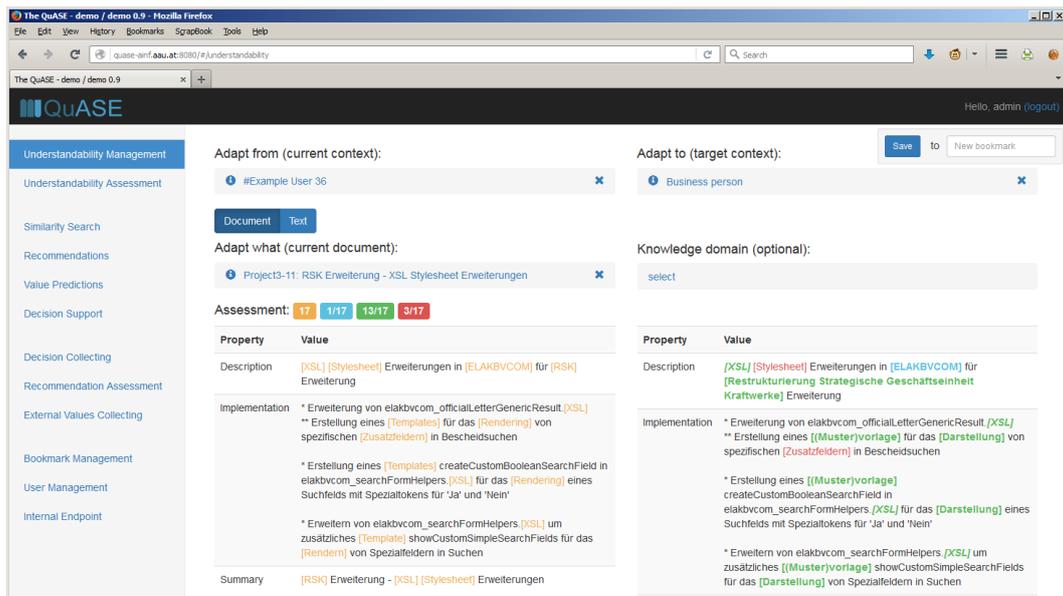


Fig. 14. Document translation screen

On the translation screen, the source document and the translation results are shown in two columns alongside each other. Different terms in the document are highlighted as follows:

1. The terms in the source document which are the candidates for adaptation are shown in orange, if the corresponding translation is selected in the translated document, the element is highlighted in yellow (see Fig. 15).
2. The terms in the translated document which have been translated and/or explained successfully (the target context has corresponding meaning element with different representation) are highlighted in green.
 - 2.1. If only the explanation is available (i.e. the target context contains the corresponding meaning element with the same representation but different explanation) the green-highlighted term is additionally shown in italic.
3. The terms in the translated document which do not need to be translated (the target context contains the same meaning element with the same representation and explanation) are highlighted in blue.
4. The terms in the translated document which cannot be translated (the target context does not contain the corresponding meaning element) are highlighted in red.

Above the source document, the statistic about this document is shown: the orange box contains the number of candidate terms, other boxes correspond to the number of the highlighted terms of the particular color.

After selecting the term in the translated document, if the explanation text has been defined for the selected term, the explanation pop-up window is displayed (this is shown on Fig. 15). In this window, the original term is shown at the top, and the explanation text below it.

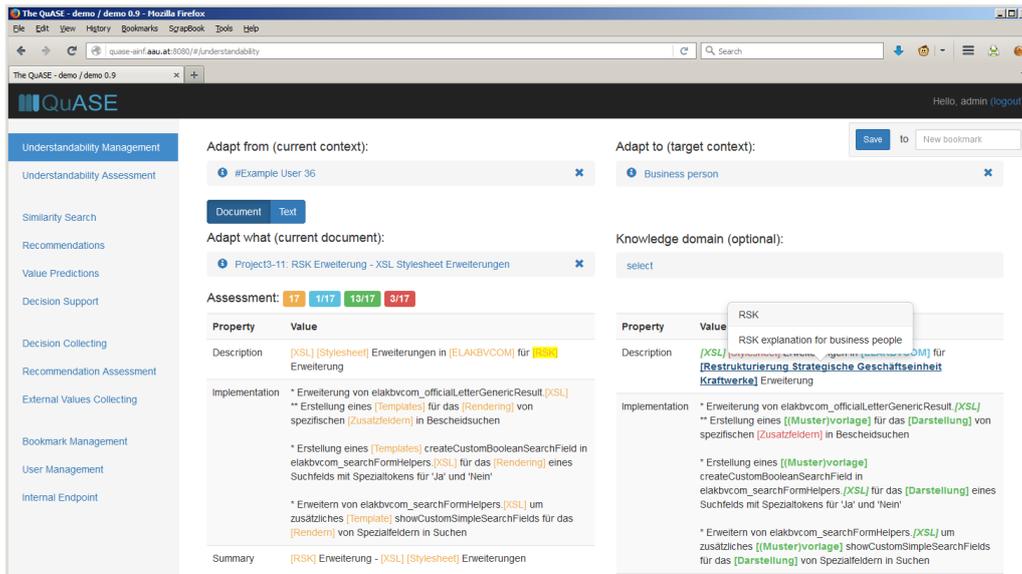


Fig. 15. Document translation screen: showing the explanation

3.4 Specifying the knowledge domain

The number of terms to be translated/explained can be further narrowed by specifying the knowledge domain (Fig. 16-17). Every concept has one or more knowledge domains associated with it, after selecting particular domain, only concepts belonging to this domain will be used for translation/explanation. Domain selection is optional: if the domain is not selected, all terms are considered.

To start domain selection activity, it is necessary to choose the selector below the label “Knowledge domain (optional)”. The domains are organized hierarchically in the system, so the selection of the particular domain is performed through the dialog with the structure resembling context selection dialog (Fig. 16). The only class currently available in the class selectors in this dialog is “Ontological domain” (selected by default) so these selectors can be ignored and only element selection performed.

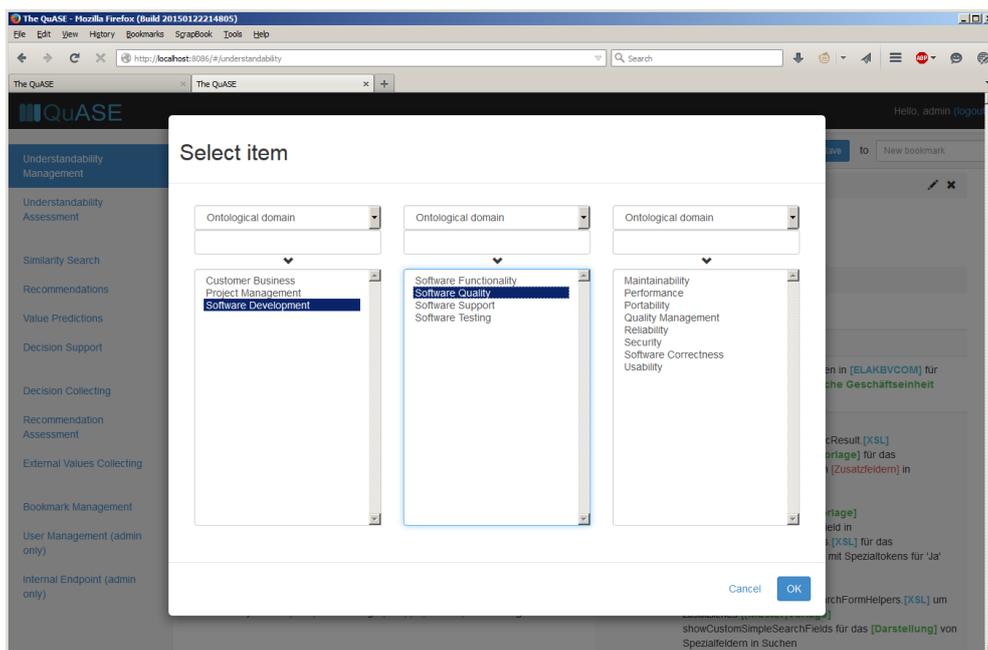


Fig. 16. Selecting the knowledge domain

After selecting the domain, the number of the translated terms will be limited to those belonging to this domain (Fig. 17, note that, as compared to Fig. 14-15, “Template” and other terms not belonging to the selected domain are not translated).

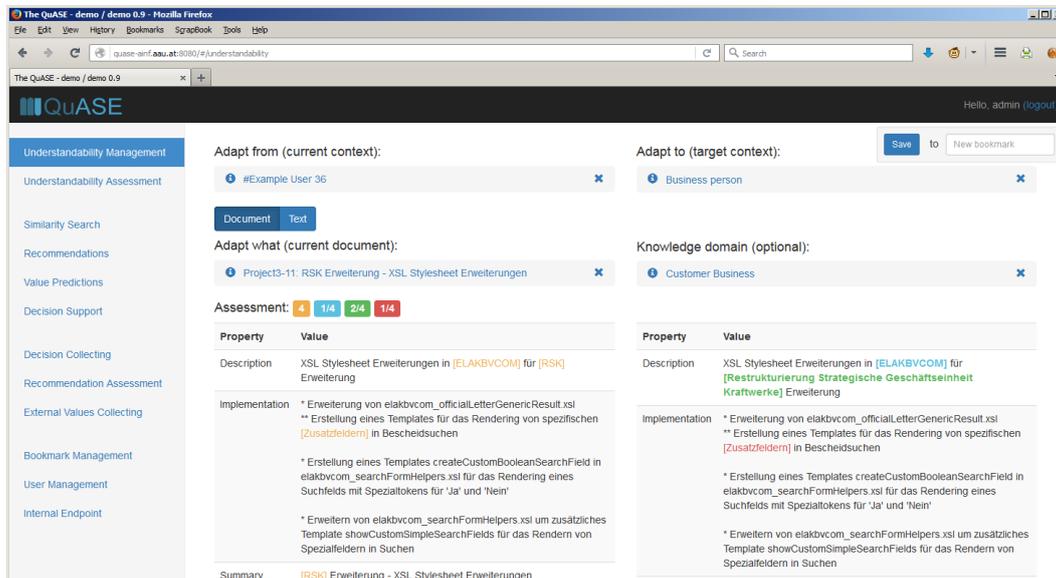


Fig. 17. Translation screen with selected knowledge domain

3.5 Translating arbitrary text

Besides working with documents, it is also possible to improve the understandability of the arbitrary text specified (e.g. typed or pasted) by the user (Fig. 18). To do so, it is necessary to select “Text” in the two-position “Document|Text” switch below the source context selector. After this, the input area for specifying the text becomes available together with context and knowledge domain selectors functioning in the same way as has been described for the case of document selection. The translation is performed by clicking on the “Translate” button below the area.

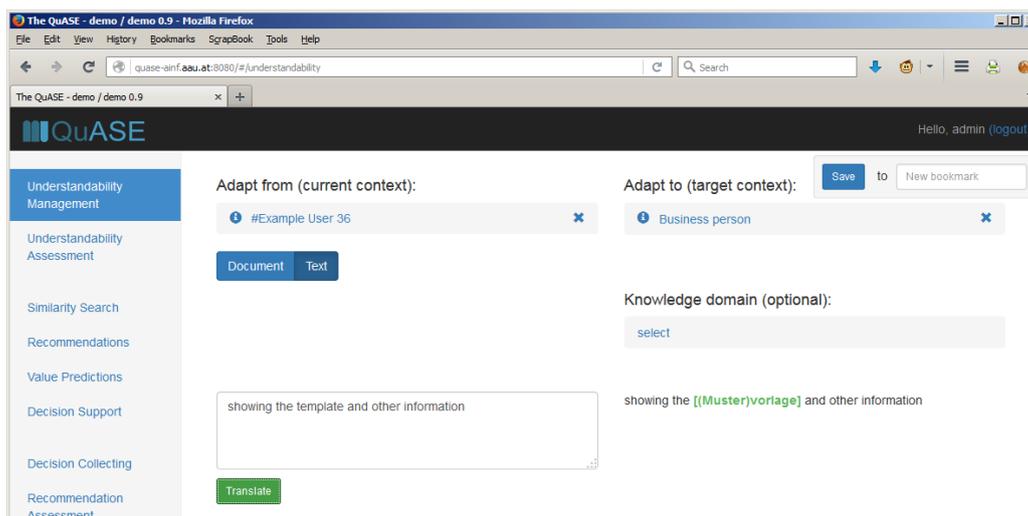


Fig. 18. Translating arbitrary text

3.6 Understandability assessment

The purpose of understandability assessment is to allow the user to select the arbitrary number of documents and observe understandability-related characteristics of these documents in a list. This activity is launched by selecting “Understandability assessment” in the main menu.

The first difference from the improvement UI is that the document selector allows to select more than one document; the corresponding list is multi-select (Fig. 19);

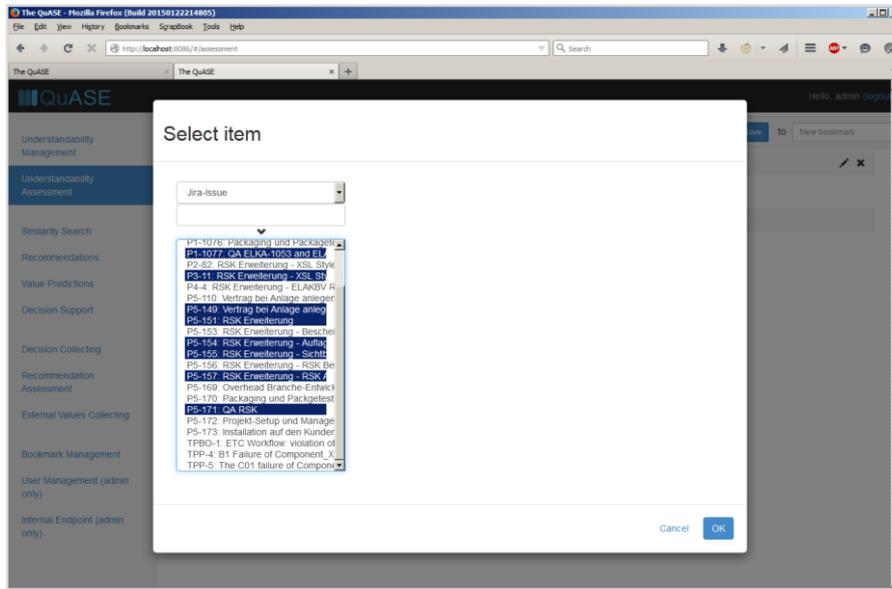


Fig. 19. Understandability assessment: selecting documents

After selecting the documents they are displayed in a list, for every document in this list (Fig. 20), the values of the understandability-related characteristics (currently, the document statistics described above) are displayed, the list itself is ordered by descending total number of the candidate terms; the documents with no candidate terms are not shown.

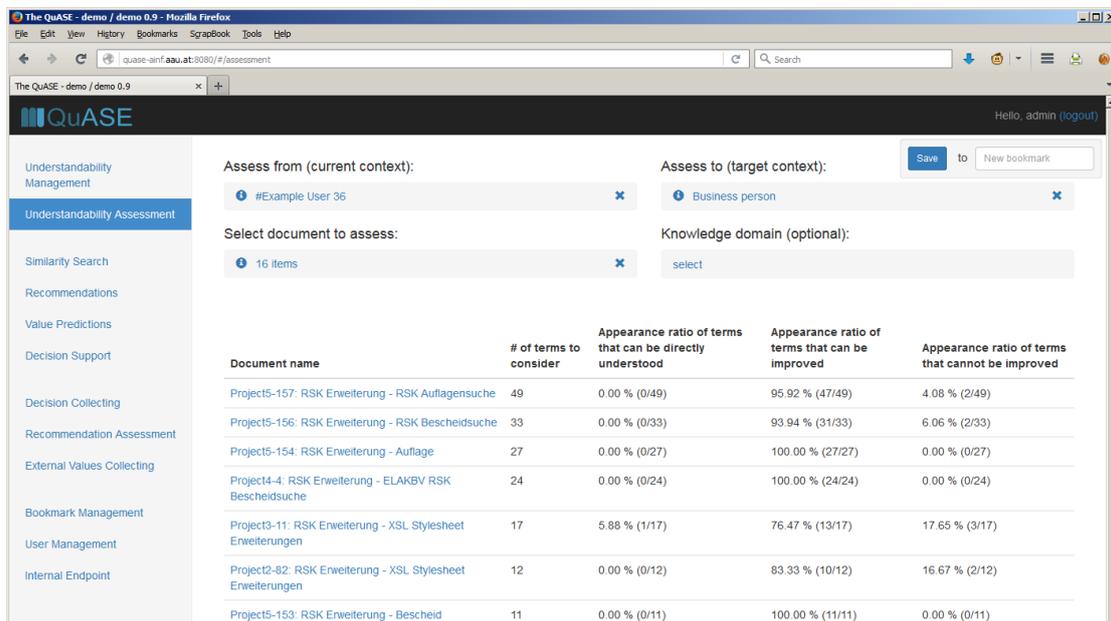


Fig. 20. Understandability assessment: displaying the results

The title of the document in the list is a link, following such link leads to displaying the pop-up window with the translation of this document (Fig. 21).

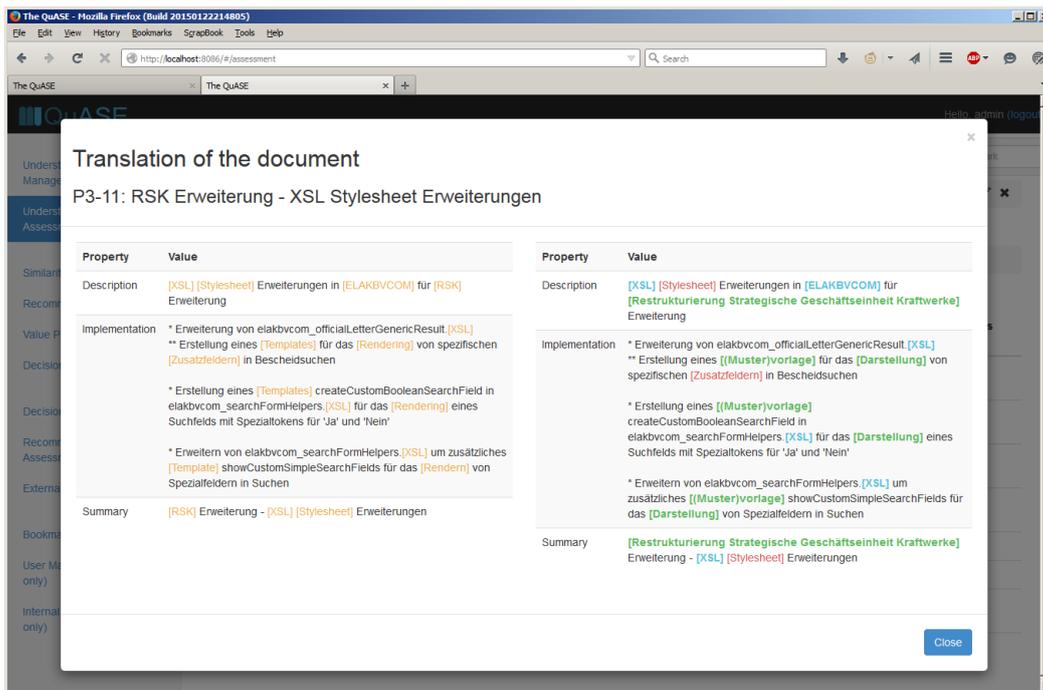


Fig. 21. Translation screen accessed from understandability assessment results

4 Analysis

4.1 Collecting external information

Collecting external information is the preliminary activity for performing the similarity search and value predictions (see below).

External information in QuASE corresponds to the values of the metrics and the entities which are specified as to be supplied by the user. QuASE allows to collect the following categories of the external information (the configuration of this information is specified in the site model):

1. Custom metric values related to repository-based entities (such as Jira issues, developers, or projects)
2. Custom metric values characterizing the relations between repository-based entities (such as developer attitude to the issue or developer degree of involvement into project)
3. Custom entities (not existing in the source repository) such as business customers or customer organizations for configurations where they are not collected in Jira, these entities could be supplemented by metrics
4. Relations between custom and repository-based entities (such as the relations between issues and involved business users) and between different custom entities (e.g. between business users and business organizations), custom metrics characterizing these relations (such as customer attitude to the project).

To specify external property values and entities, it is necessary to select “External values collecting” in the main menu. After that, it is necessary to select the element for which the values have to be collected (the *source unit*, the available elements are those for which the external attributes are specified in the site model). After selecting this element, the input elements for specifying the values become accessible (Fig. 22). Note that the “Save” button becomes enabled only if there were any changes in the values and becomes disabled again after saving the values.

Possible types of attributes include direct input (which allow for entering the value directly), and two kinds of attributes whose values have to be selected from the list: “level” (with values ranking from

“very low” to “very high”) and “attitude” (with values ranking from “very negative” to “very positive”), on Fig. 22 the attributes are of the “level” type.

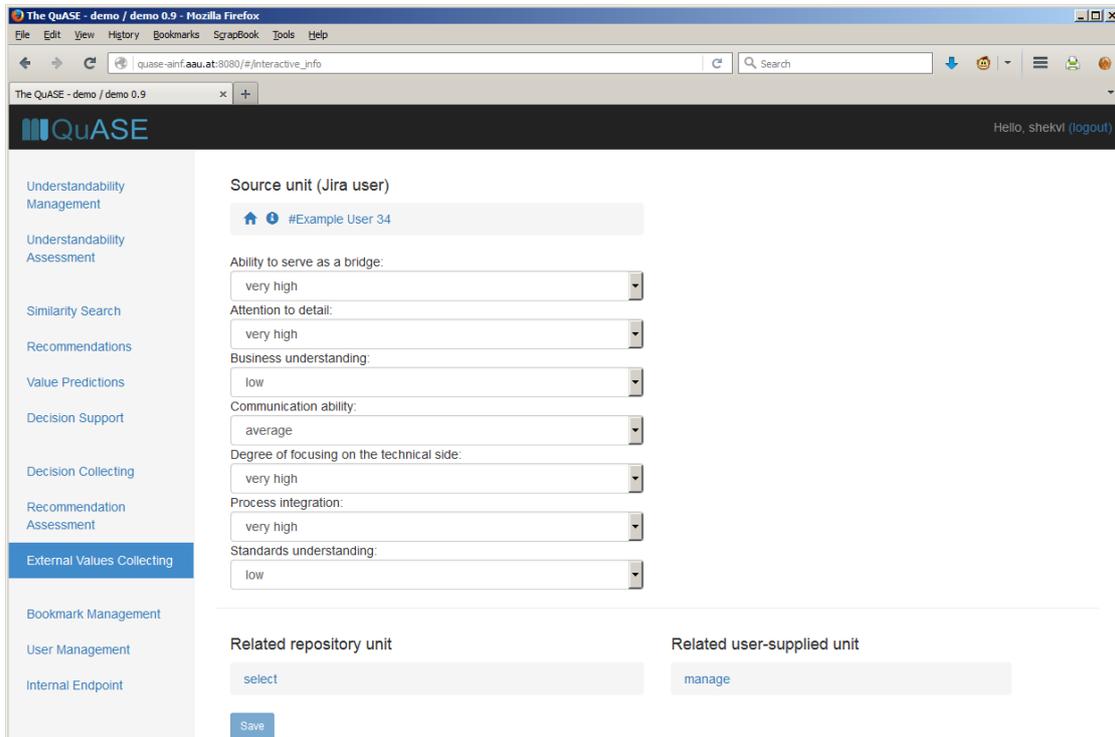


Fig. 22. Specifying extra information for repository-based unit (Jira user)

Below the list of the custom metrics for the selected unit, one can see a “Related repository unit” selector which allows specifying the related repository unit with a purpose of supplying the values for the metrics which characterize the connection between the selected unit and the related unit. The corresponding selection screen is shown on Fig. 23, the visible units are those connected to the source unit with relations accompanied by metrics (on Fig. 23, the issues belonging to the previously selected Jira user are shown because the site model defines the metrics for the relation between Jira users and issues).

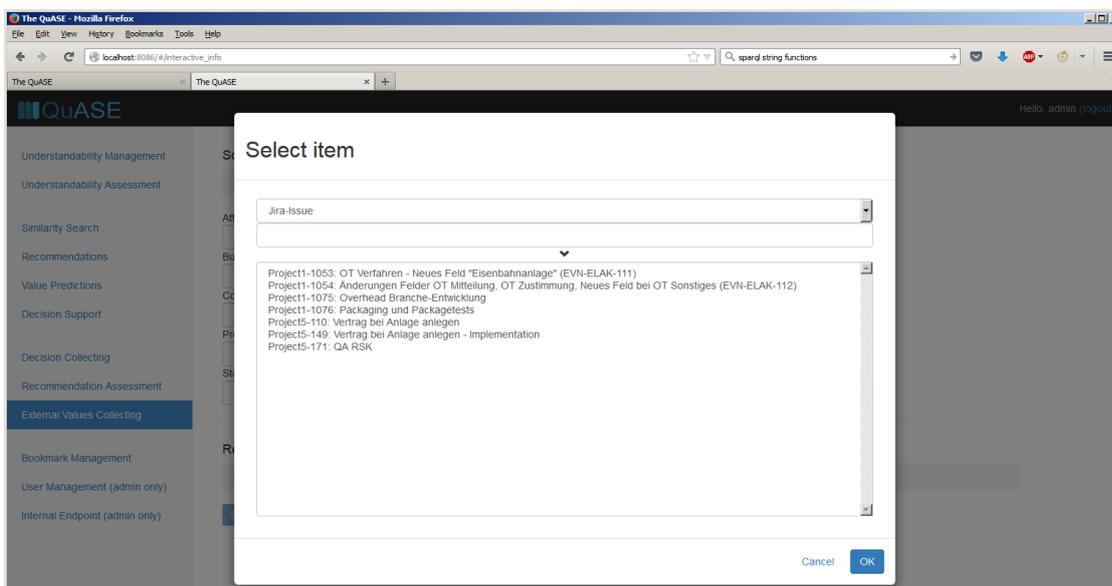


Fig. 23. Selecting related repository-based unit (Jira issue authored by selected Jira user)

After selecting the related unit, the list of input fields organized similarly to the list for the source unit becomes visible below the selector (Fig. 24), these fields correspond to the metrics characterizing the relation between these units. Selecting different related unit makes available the set of fields to specify metrics characterizing the new connection etc. Note that the “customer attitude” metric shown on Fig. 24 is of the “attitude” type.

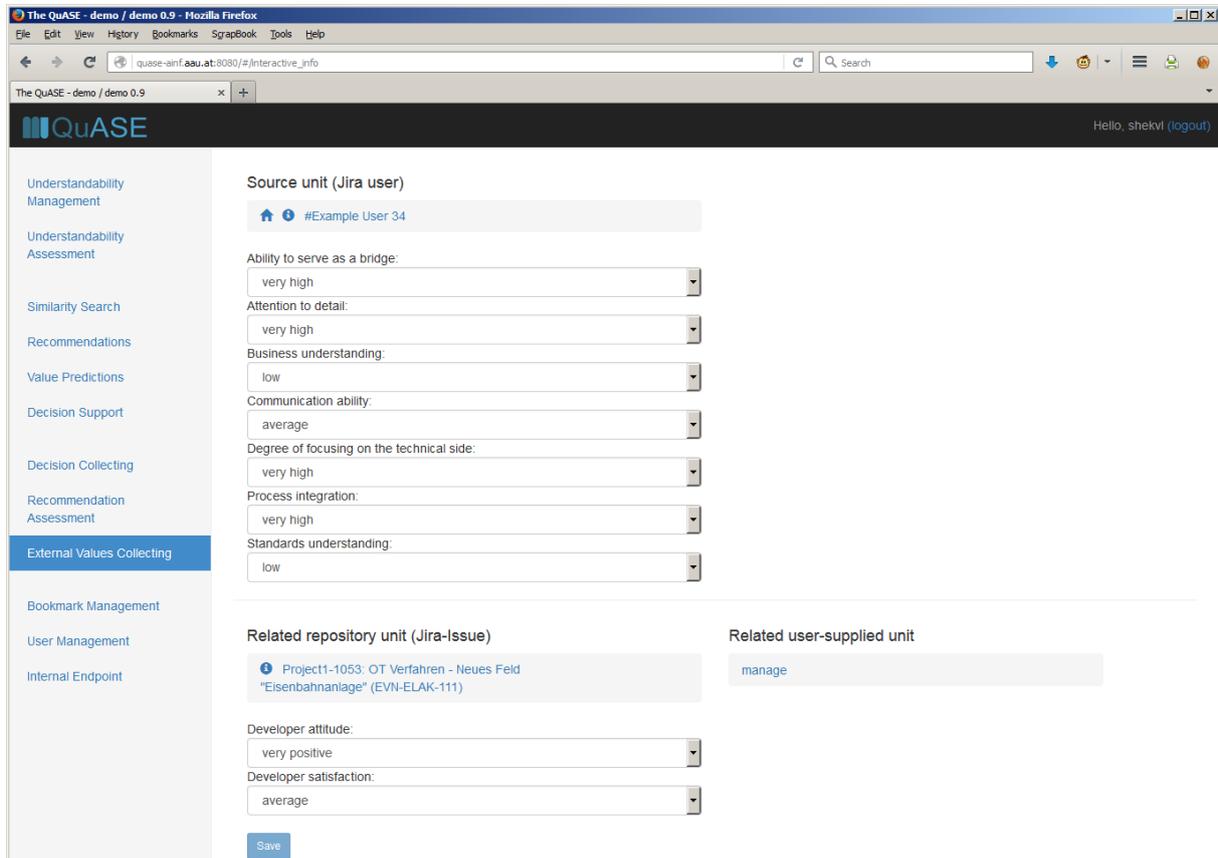


Fig. 24. Specifying metrics for the relation between repository units

Up to this point, external information specified by the user was related to the entities originated in the project repository (e.g. existing in Jira) and existing connections between them (i.e. we cannot modify the connections between Jira users and issues in QuASE, we can only specify the metrics related to these connections). In addition to that, QuASE supports the concept of *user-supplied units*, such units can be managed by the user through the interface of the QuASE tool together with the connections between such units and both repository and other custom units.

To manage the custom (user-supplied) units, it is necessary to use the selector “Related user-supplied unit” to the right of “Related repository unit”. The custom unit management window shown as a result of selecting “manage” in this selector is shown on Fig. 25.

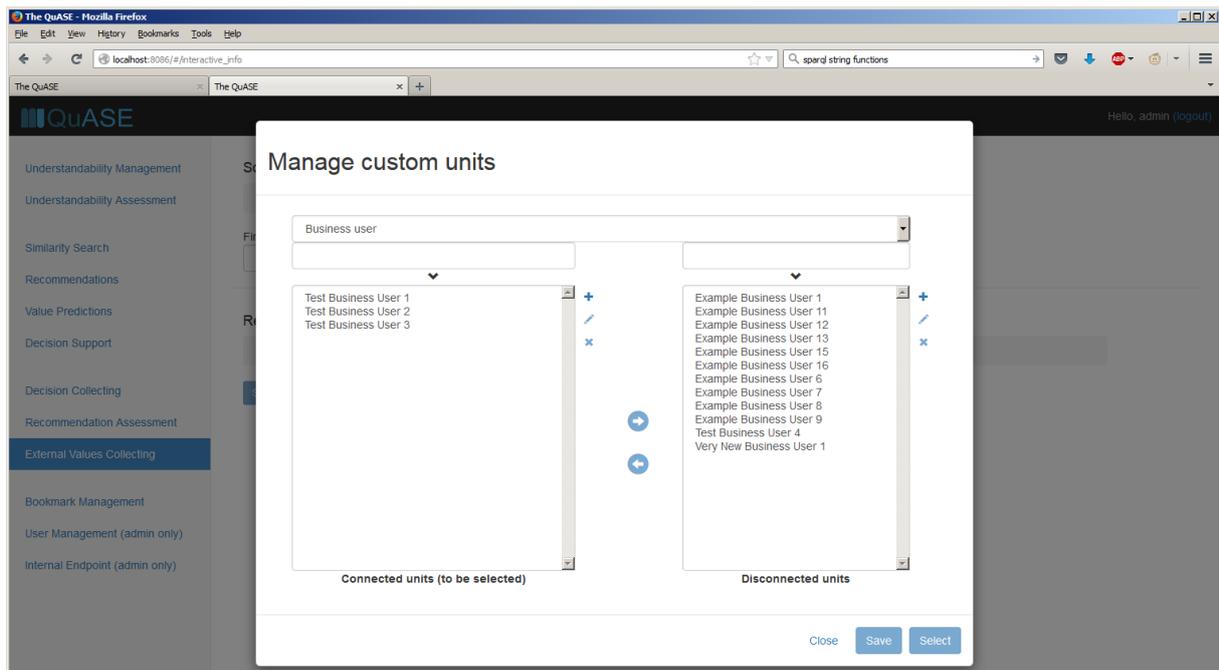


Fig. 25. Managing custom entities

Prior to continuing, it is important to note that all changes made by means of the custom unit management dialog are collected together and have to be explicitly saved by clicking on the “Save” button on the bottom right of the dialog window. This button becomes enabled after any change is made, it becomes disabled on saving (the dialog is not closed by means of this button). If the changes are not saved, selecting “Cancel” to the left of the “Save” button abandons all the changes made so far, if the changes are saved, this button will change its label to “Close” indicating that closing the window is now safe. In addition, selecting the custom unit by means of this dialog (see below) is only possible if the changes have been saved.

The custom unit management dialog contains the single class selector on the top of the screen (allowing to switch between the types of the units to be managed) and two lists below it. The left list contains the custom units of the particular type connected to the source unit, the right list – all other units of this type. Moving the units between these lists by means of the arrow buttons residing between the lists creates or deletes connections between custom units and the source unit. Multiple units can be selected for moving in both lists.

For managing the units, three editing buttons are available to the right for both left and right lists. Selecting the top button leads to displaying additional creation dialog (Fig. 26) where all properties of the unit to be created can be specified (note that these properties not only include the metrics as was in the case for repository-based units, e.g. on Fig. 26, the “full name” property can be specified for a new business user). Creating the unit in the left list automatically specifies a connection between this new unit and the source unit.

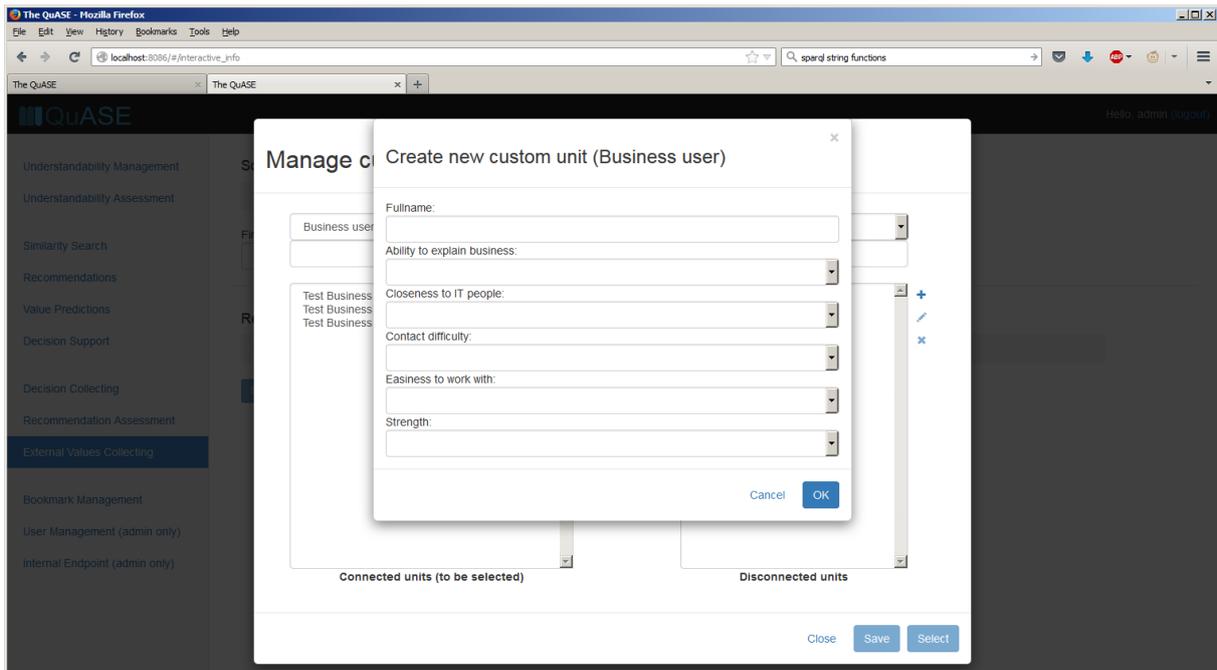


Fig. 26. Creating new custom unit

Other two buttons perform actions on the unit selected in the corresponding list (they become enabled only if the unit is selected). Selecting the middle “edit” button leads to displaying the editing dialog (Fig. 27) organized similarly to the creation dialog, selecting the bottom “delete” button deletes the unit, on saving the changes, this unit will be deleted from the knowledge base and will not appear in any of the lists anymore (not only its connection will be deleted).

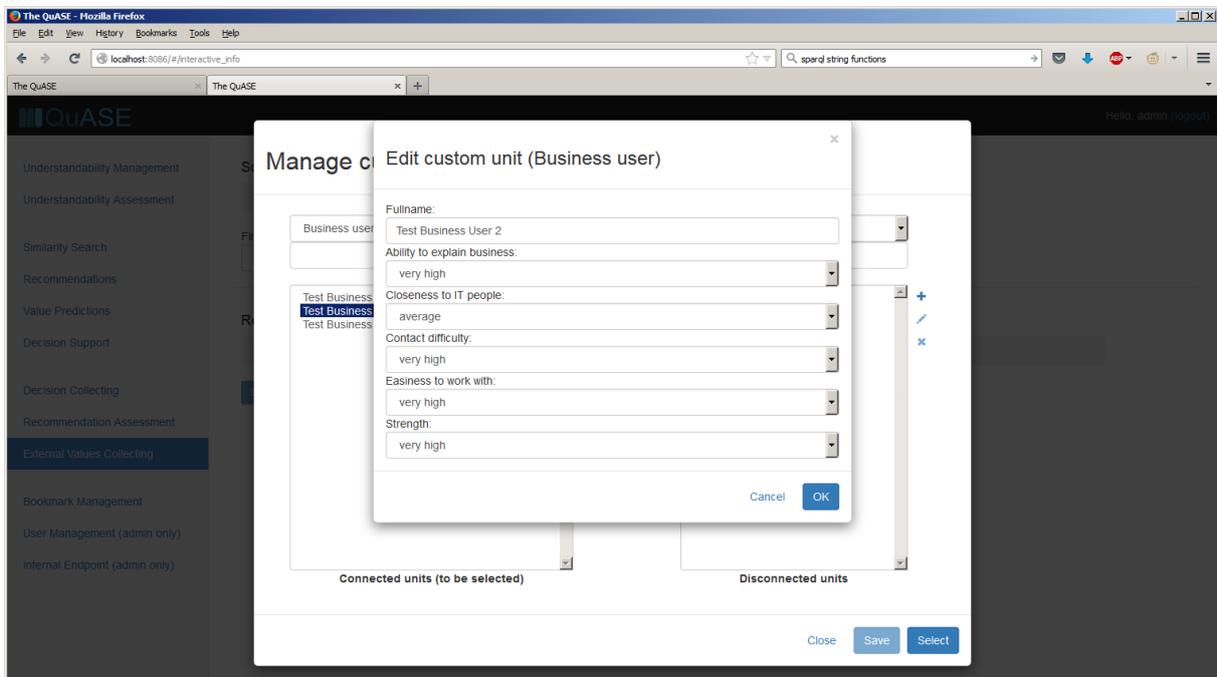


Fig. 27. Modifying existing custom unit

If there are no pending changes in the dialog, it is possible to select the connected custom unit via the management dialog in the same way as the related repository unit has been selected (see Fig.20 and the accompanying text). The item is first selected in the left dialog, then the “Select” button

becomes enabled (Fig. 28): this button does not become enabled if there are pending changes or there are no single-item selection in the left list.

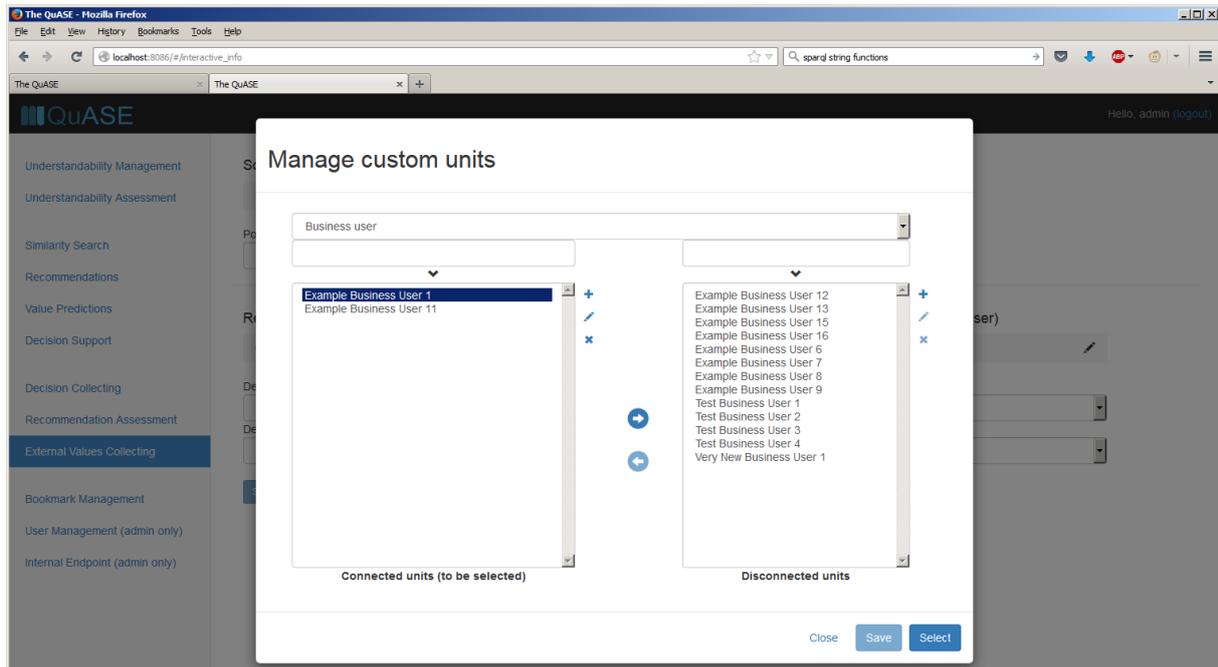


Fig. 28. Selecting connected custom unit to specify metrics for its relation to the source unit

Clicking on the “Select” button closes the management dialog and makes possible to specify the values of the connection between the selected custom unit and the source unit (Fig. 29). This is implemented exactly in the same way as for the connection between the source unit and the repository unit (see above).

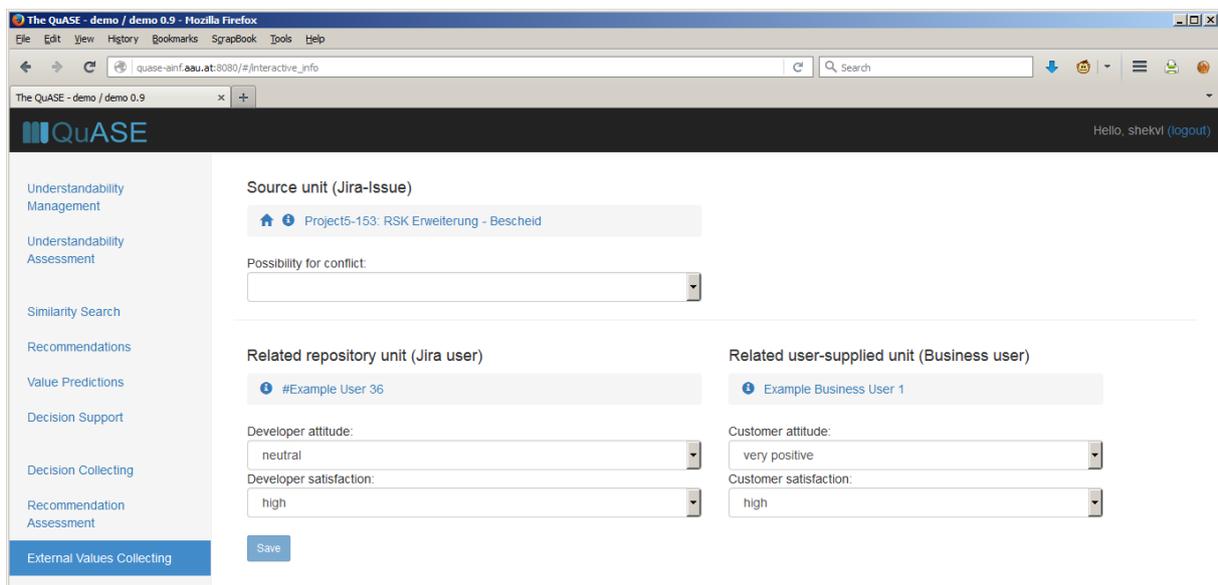


Fig. 29. Specifying metrics for the relation between repository unit and the custom unit

4.2 Aggregate metrics

Every numerical property (metric) in QuASE can be transparently aggregated to form aggregate metric (such as e.g. average customer satisfaction with issue or minimum degree of involvement of

the customer to the project). These aggregate metrics are specified in the QuASE site model and do not require special maintenance, they can be used for analysis in the same way as directly specified metrics. Every change of the source metric leads to immediate background recalculation of all dependent aggregate metrics.

Fig. 30 shows the unit information screen (displayed after clicking on the name of the element in any selector) including the values for “invisible” aggregate metrics.

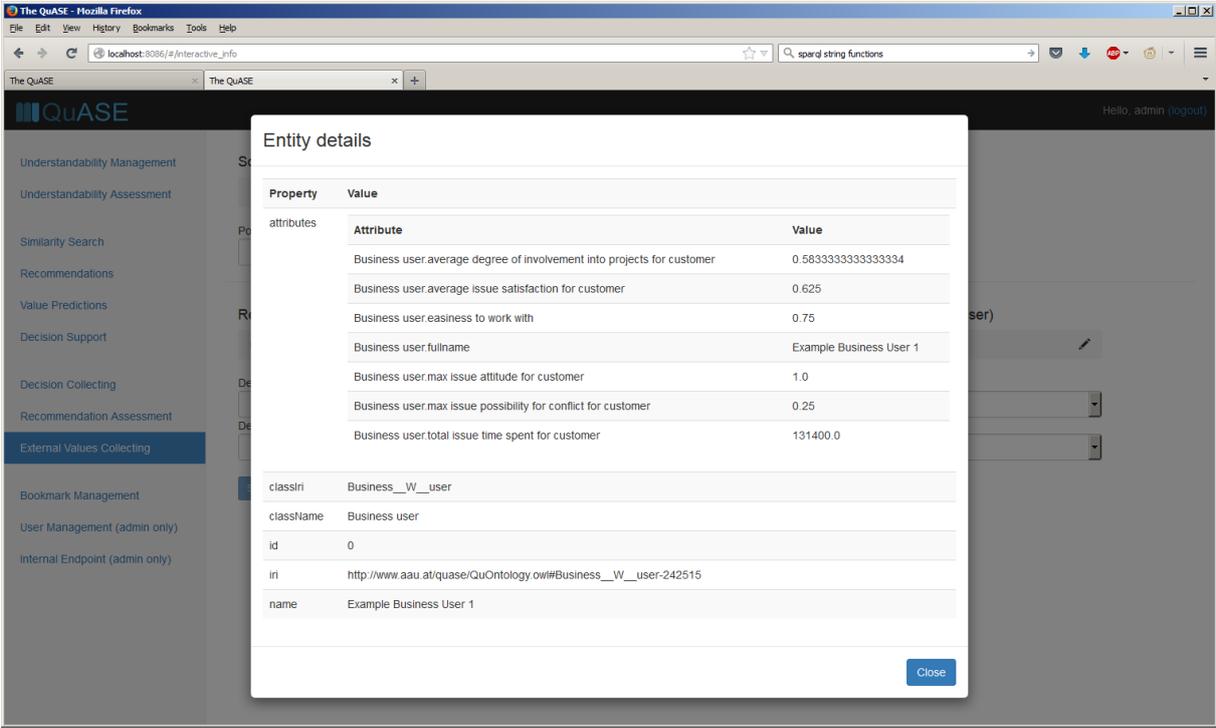


Fig. 30. Unit information screen with the metric values (including aggregate metrics) shown

Metrics for relations cannot be directly used for analysis, they can only serve as sources for aggregate metrics.

4.3 Similarity search

Similarity search is used to find the documents or context elements which are similar to the selected element based on selected number of metrics. To perform such search, it is necessary to select “Similarity search” in the main menu.

Similarity search activity is started by selecting the similarity search target through the only visible selector. This selector is connected to the selection dialog (Fig. 31) where the class selector allows to select the class of the element (this selector include all classes for which the site model allows similarity search, both content and context units can be included here). Selected individual becomes a target for the search.

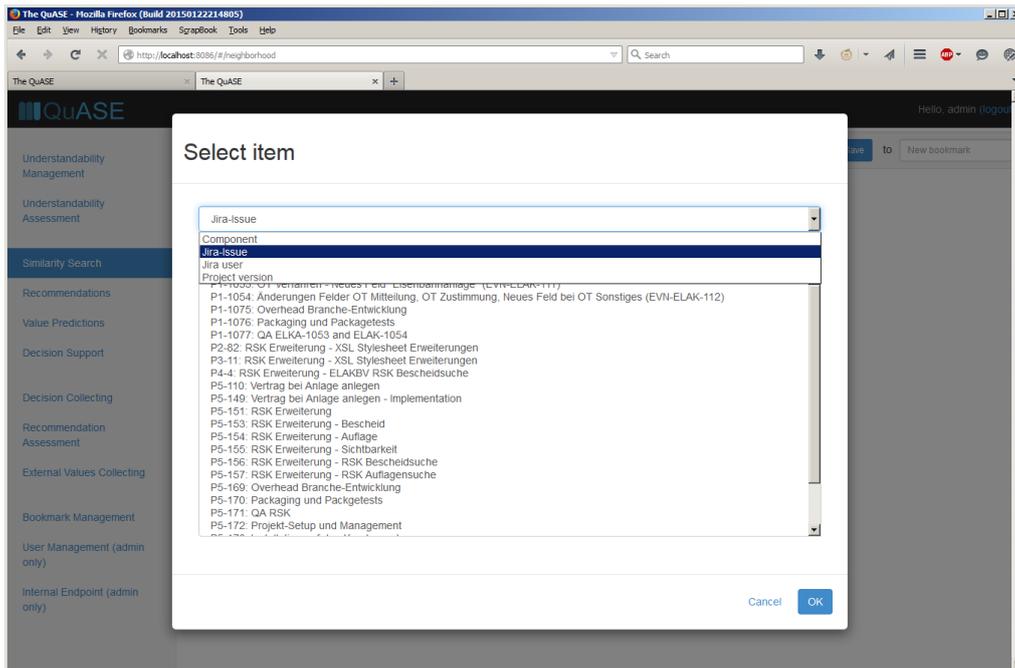


Fig. 31. Selecting target individual for similarity search

After the target is selected, the next step is to select the subset of metrics which will be used to calculate the similarity (Fig. 32). The corresponding selection list allows multiple selection, to complete the selection, it is necessary to click outside the selection window.

It is important to know that the only metrics which are visible in this list are the metrics with the values defined for the search target (the metrics which are undefined for the search target will be excluded from the list, particular example of such metric is the user-defined external metric (see Section 6.2) for which no value has been specified by the user).

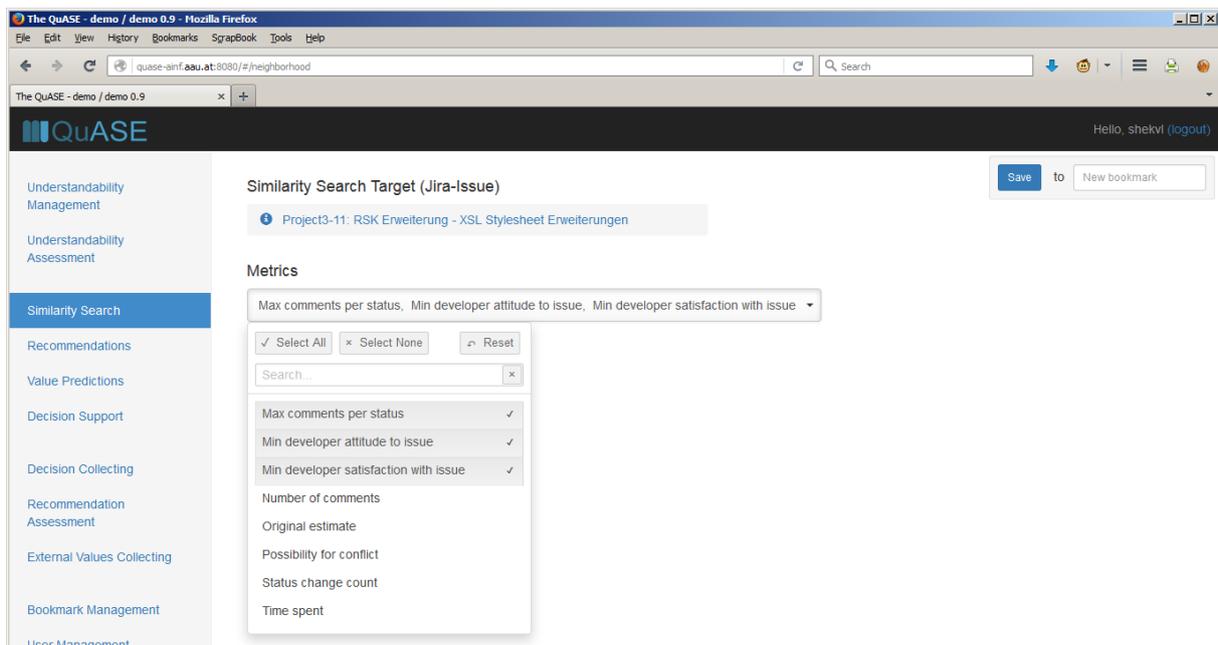


Fig. 32. Selecting metrics for similarity search

After the metrics are selected (Fig. 33), clicking on “Get similar entities” button performs a similarity search. For this search, the distance between the target and the entities of the same type in the KB is calculated, and the set of entities is returned sorted by distance in descending order.

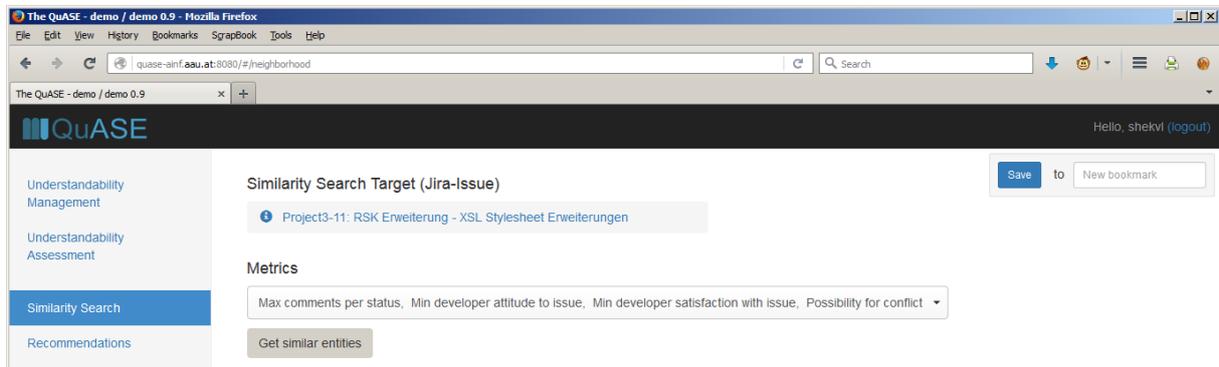


Fig. 33. Similarity search screen with selected target and metrics

The example of the search results is shown on Fig. 34. The list columns are the rank of the list item, the name, the values of the specified metrics, and the distance (titled “Similarity”). The list starts from the target entity (titled “Input entity” ranked 0, the similar entities are shown below “Similar entities” label with ranks starting from 1. The names in the list are clickable, selecting such name leads to displaying the pop-up window with the additional information about the individual.

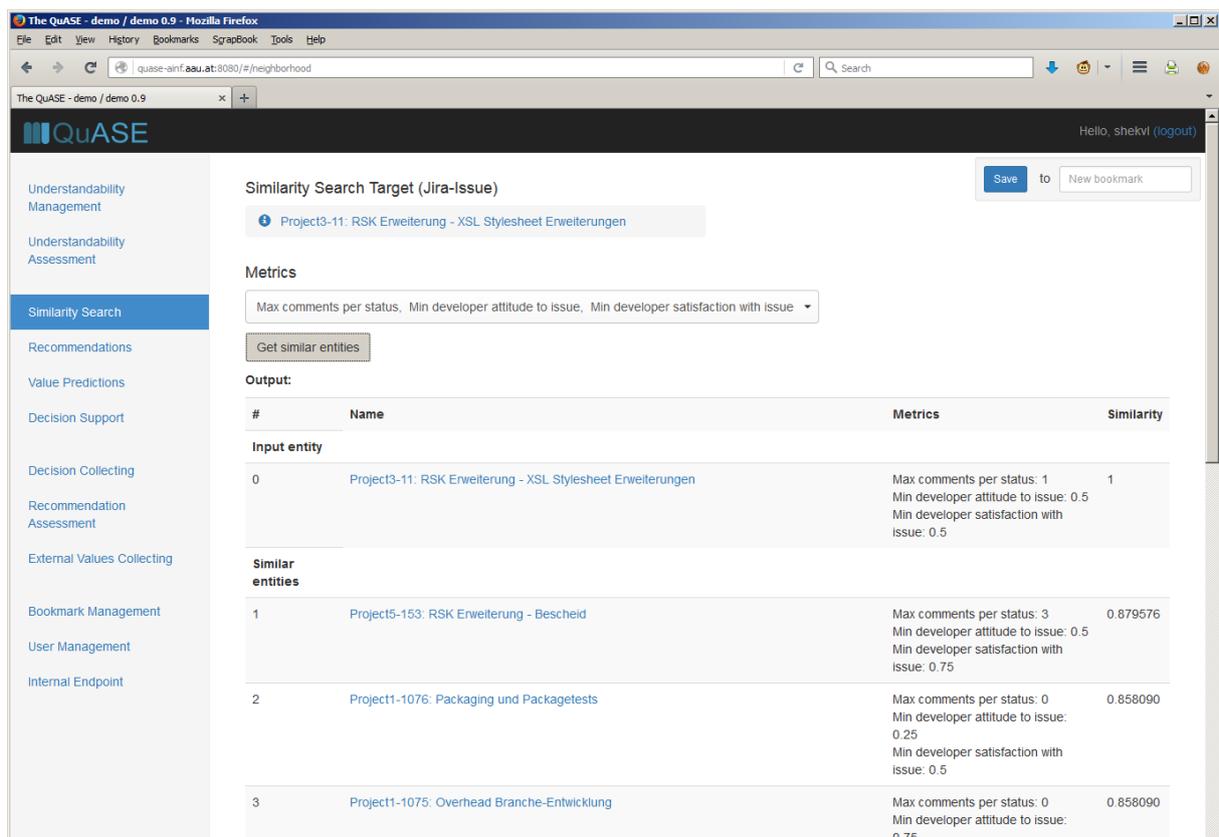


Fig. 34. Similarity search: displaying results

4.4 Predicting metric values

Metric values for the particular elements can be predicted based on the values belonging to similar elements. The necessary actions are similar to those performed for similarity search (Fig. 35): to get predictions, it is necessary to select prediction target and a set of metrics to be predicted, the

difference in this case is that the metrics whose values are not defined for the prediction target are still available for prediction. The results are displayed in a list.

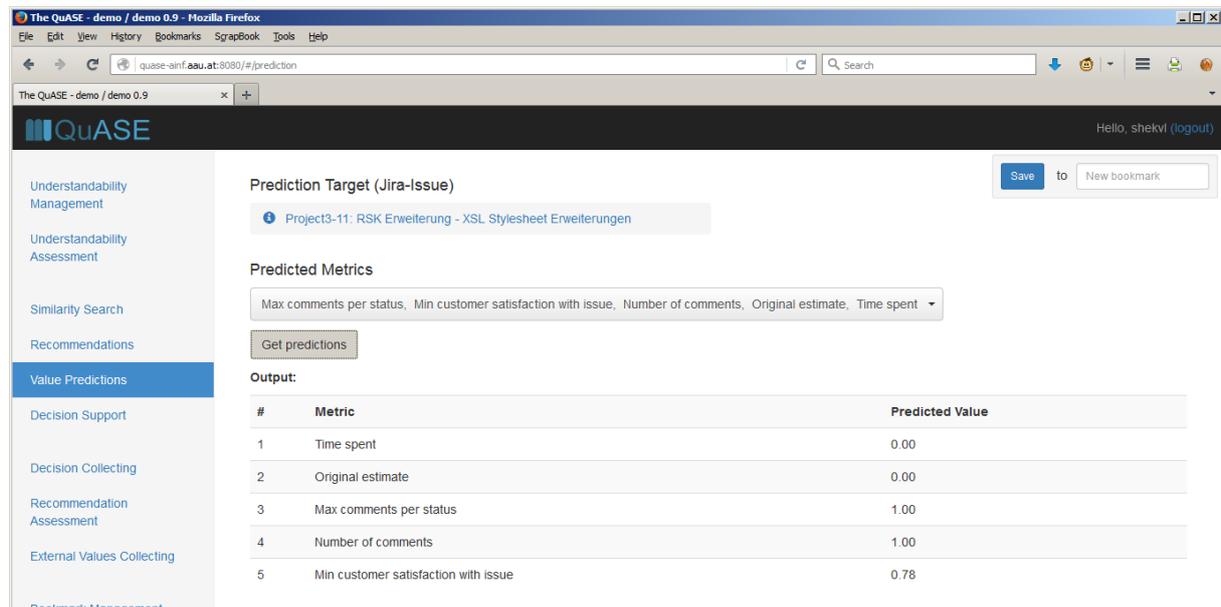


Fig. 35. Displaying the prediction results

4.5 Collecting decisions and recommendations

Using decision support and recommendations functionality requires performing the same preliminary sequence of actions for collecting information with the following differences:

1. The recommendation activities address the future actions (some of the alternatives for action is recommended for the context element or the document), so the alternatives for recommendations have to be preliminarily assessed by QuASE users as acceptable or unacceptable, their assessments are captured and used to fulfil subsequent requests.
2. The decisions support activities deal with the decisions performed in the past, past selected alternatives have to be preliminarily collected and are used to predict the outcomes of the decisions performed in future.

In this section we restrict ourselves with the decision support activities, the recommendation support activities are performed in the similar way.

Prior to specifying the decision (Fig. 36), it is necessary to select the element serving as a decision target (via a selector with a label "Select unit"): the element for which the decision is going to be collected, and a decision assessor (via "Select assessor" selector): a context unit, usually a person, whose decision is going to be collected; for the users connected to the home context, the assessor will be preselected to be equal to this context as on Fig. 36. After this, in the left column, the list of the available decision kinds (the types of decisions connected to the class of the selected target) becomes available, every kind is accompanied by the list of alternatives, one alternative could be selected from this list. The right column shows the list of decisions already collected for this target (its "decision history").

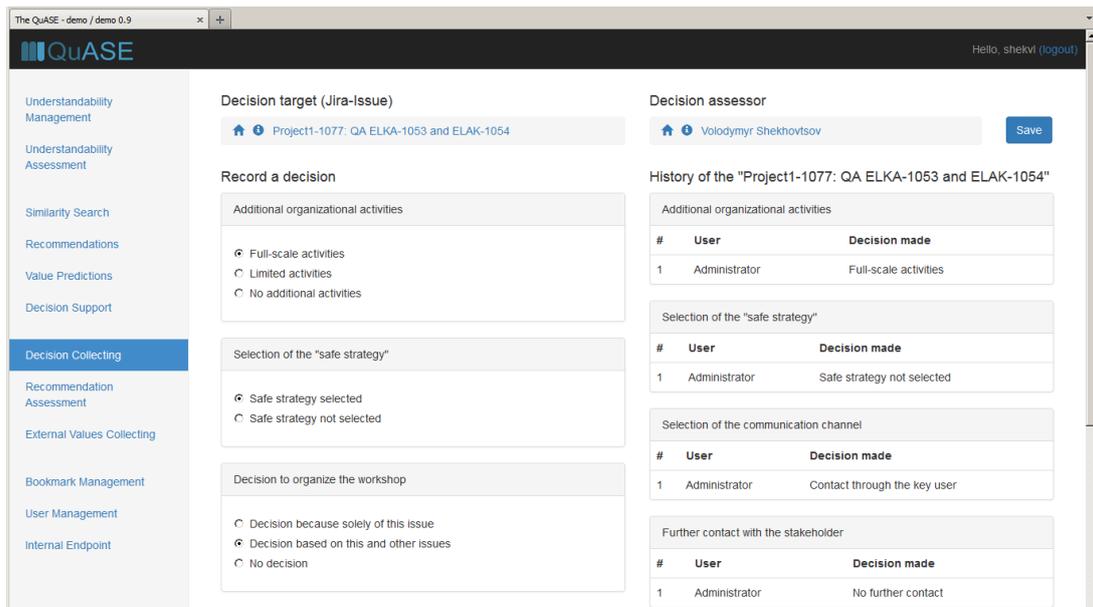


Fig. 36. Specifying the decisions

After the alternatives are selected, this selection has to be saved by clicking on the “Save” button on the top right corner of the screen. After that, the newly collected decisions will be added to the history in the right column (Fig. 37).

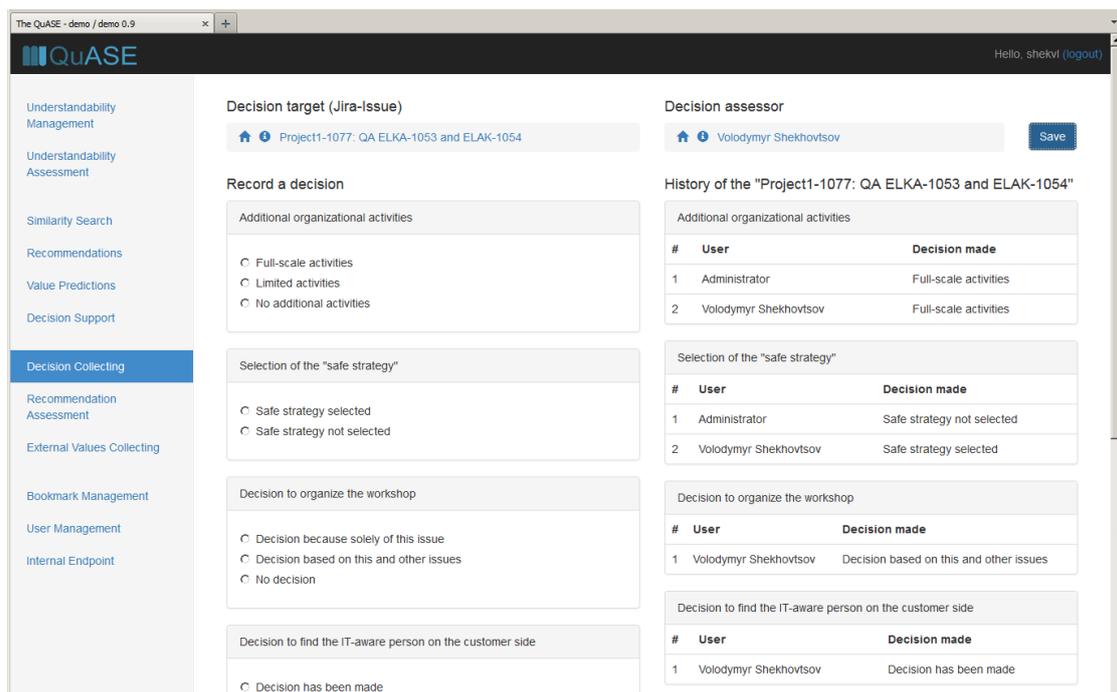


Fig. 37. Decision collecting screen with updated information

4.6 Predicting decisions and issuing recommendations

In this section, again, only decision prediction will be described, issuing the recommendations is performed similarly.

Predicting the decision starts from selecting the decision support target (the element, for which the outcome of the decision is going to be predicted). To do so, it is necessary to use the only selector initially visible. After the decision target is selected, the selector for the decision kind becomes

accessible (Fig. 38). The available decision kinds depend on the selected decision target. The decision kind determines the type of the decision to be predicted.

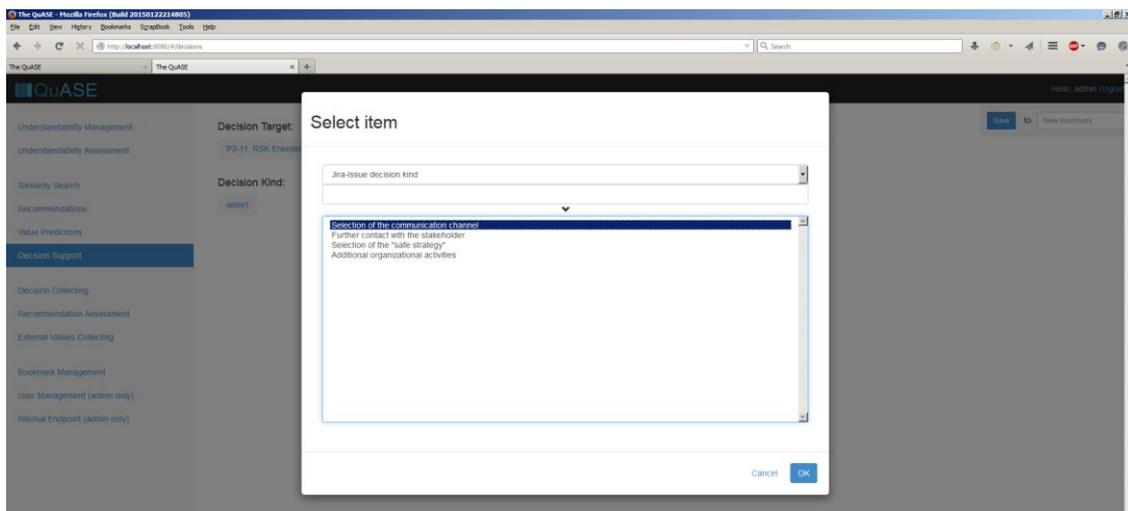


Fig. 38. Selecting the decision kind

After selecting the decision kind, the button “Get decisions” becomes accessible, decision prediction results are shown on Fig. 39, the columns reflect the rank of the alternative, its name, and the relative certainty of the prediction.

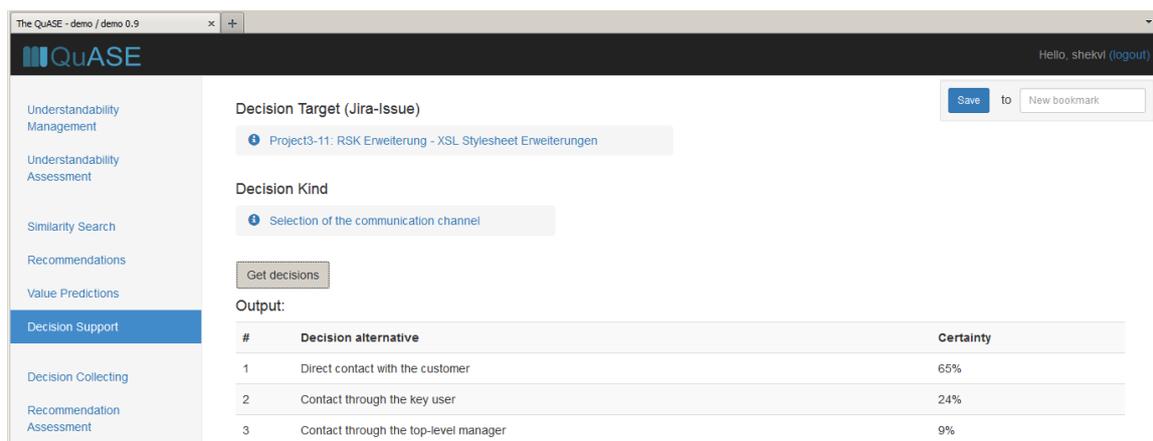


Fig. 39. Displaying the decision results

5 Additional functions

In this section, additional functions not directly related to the main functionality of the tool will be described.

5.1 Bookmark management

For every activity except specifying the extra information (section 5.1) and collecting the decision and recommendation data (section 5.4), the activity type and its input parameters (such as the selected contexts, documents, decision targets etc.) can be saved as bookmarks by using the input element in the top right corner of the screen (Fig. 40).

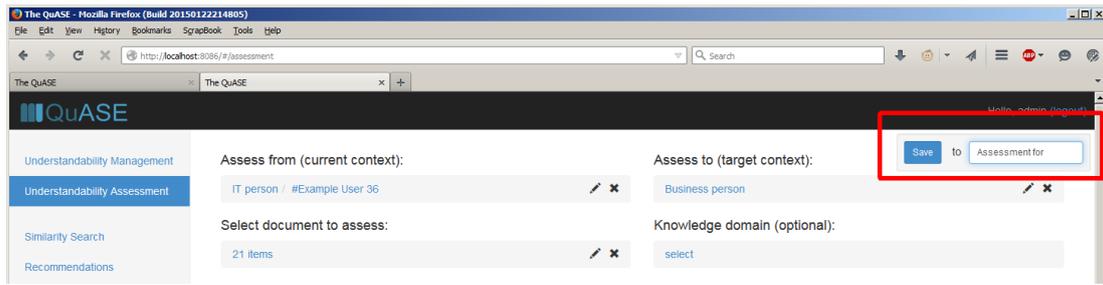


Fig. 40. Specifying the bookmark

Saved bookmarks are shown in a list accessible from the main menu (item “Bookmark management”, Fig. 41). Selecting bookmark in this list leads to the immediate navigation to the particular activity page with the values of the parameters already set.

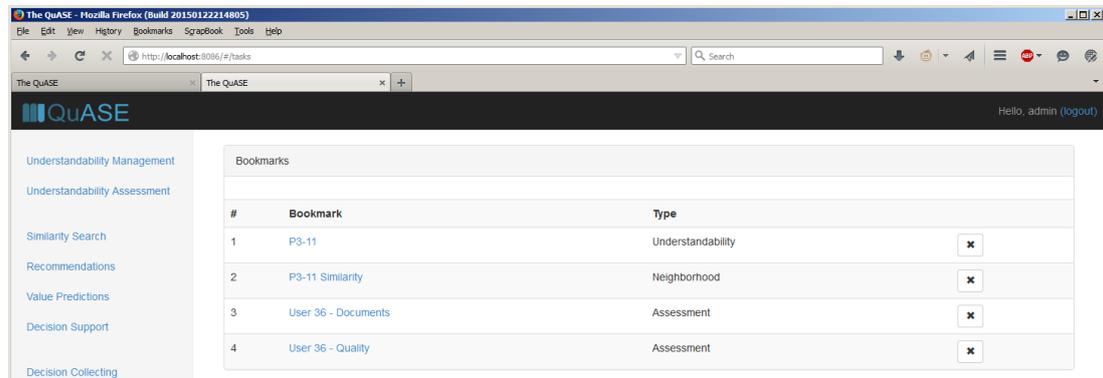


Fig. 41. The bookmark list

5.2 User management

QuASE users are managed by selecting “User management” in the main menu (Fig. 42). The list of the users is shown, with the columns corresponding to the login name, first and last names, the roles, the connected repository user, and the button for editing the information and deleting the user. It is not possible to delete the predefined Administrator account.

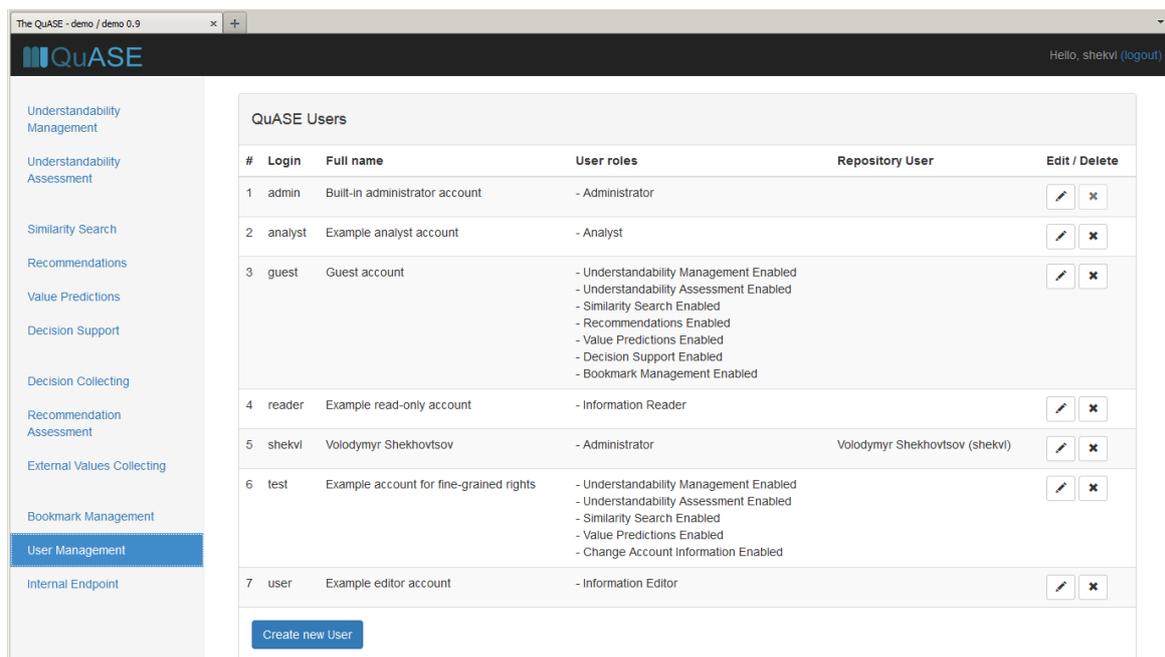


Fig. 42. The user list

Editing user information is performed in the separate window (Fig. 43).

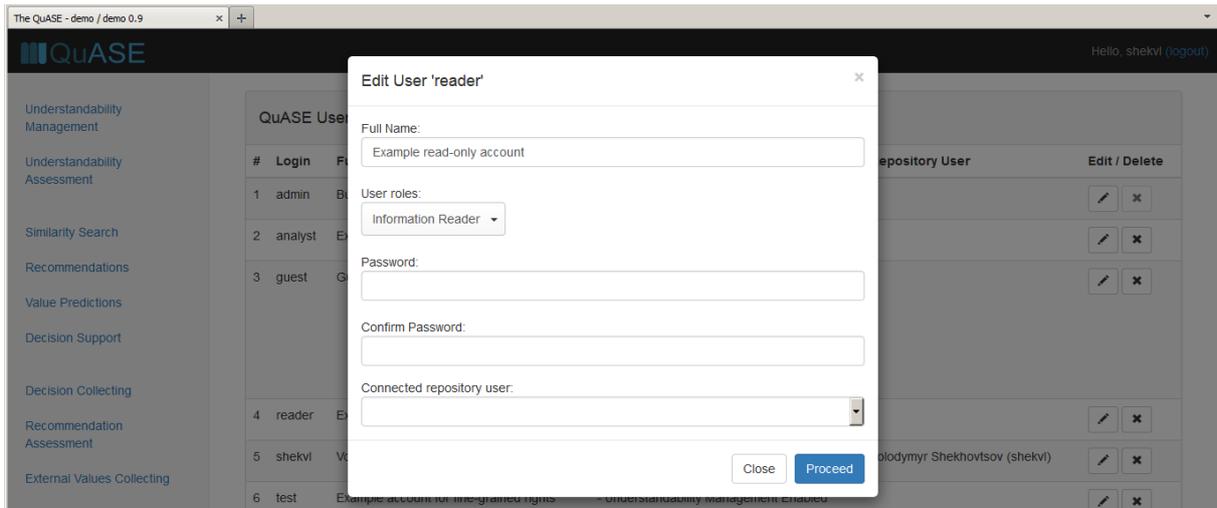


Fig. 43. Modifying user information

Roles are specified using the same multi-select input element as used for specifying metrics (Fig. 44). The detailed description of the system of roles is provided below.

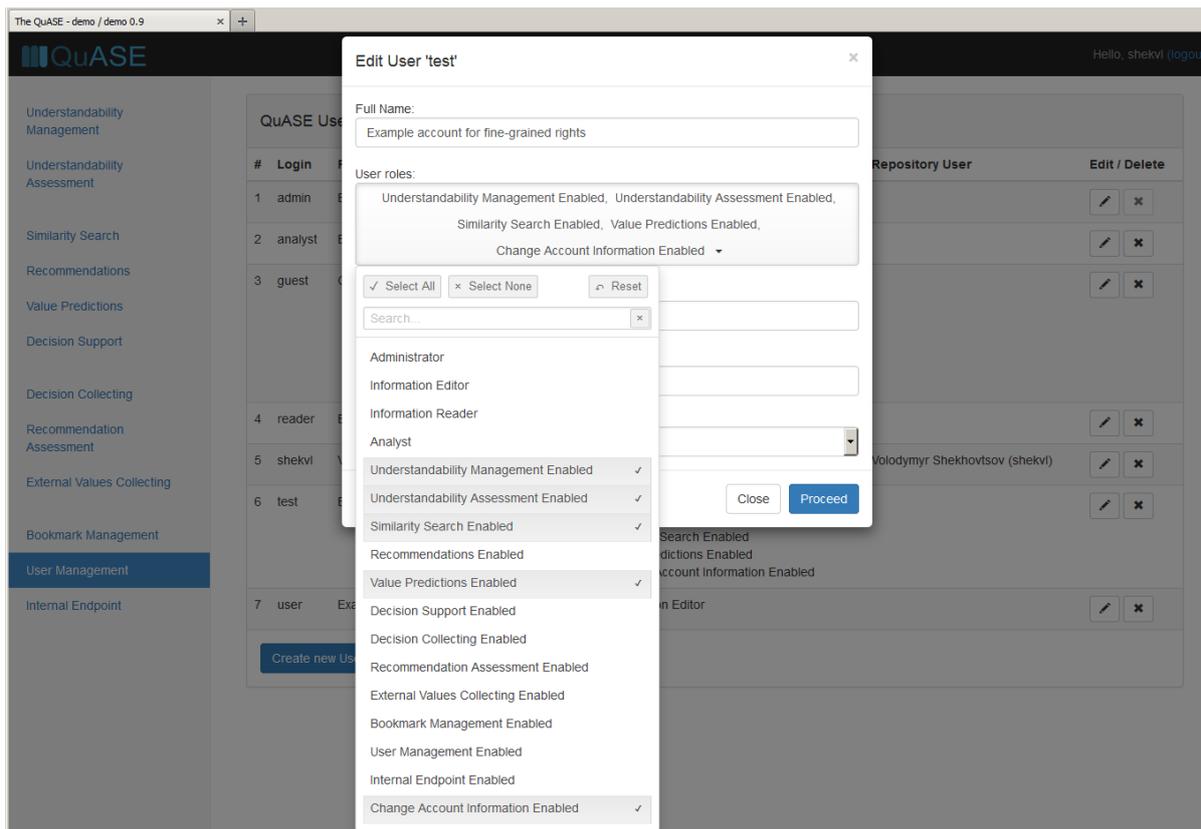


Fig. 44. Specifying roles

Connecting repository user to the QuASE user (Fig. 45) makes the corresponding home context (see section 3.1.1.) available for this user. The corresponding drop-down list allows the connected user to be selected out of the list of all Jira user individuals existing in the knowledge base.

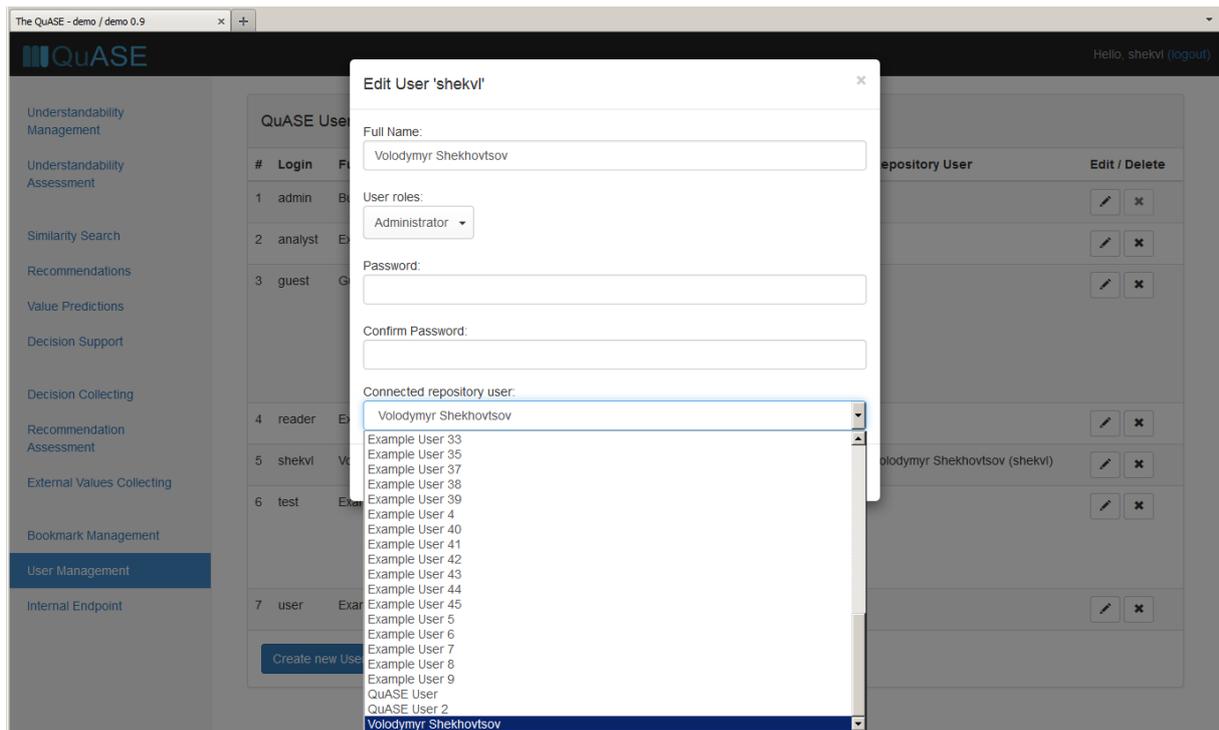


Fig. 45. Connecting repository user to the QuASE user

5.2.1 Changing account information for non-administrator users

Non-administrator users can only change the information about their own account (their full name and password as shown on Fig. 46):

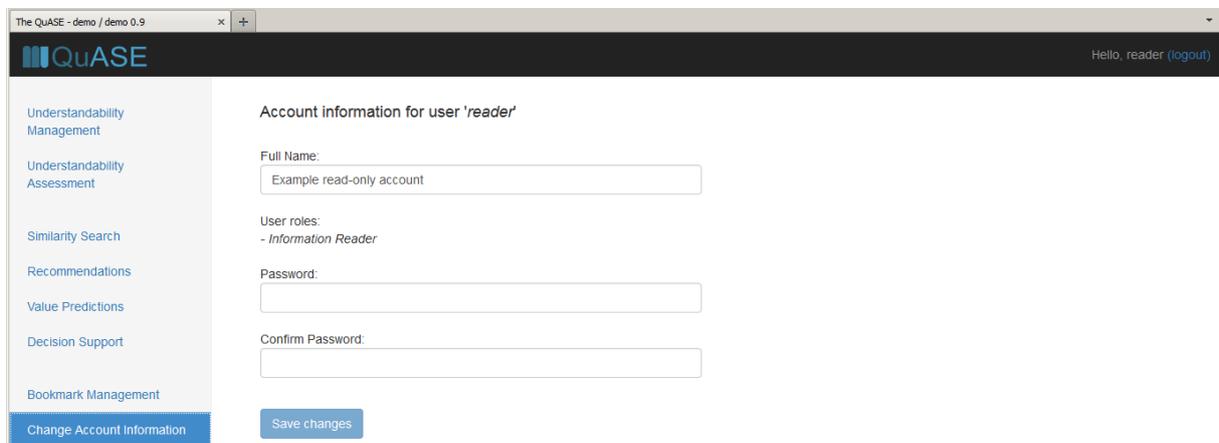


Fig. 46. Changing account information for non-administrator users

5.2.2 The QuASE role system

The QuASE role system includes the following predefined coarse-grained roles:

1. *Administrator*: can use all the functionality of the system;
2. *Information Editor*: can use all the functionality of the system except “User Management” and “Internal Endpoint”;
3. *Information Reader*: has read-only access to the system i.e. this user can use the functionality accessible for Information Editor except “Decision collecting”, “Recommendation assessment” and “External values collecting”;

4. *Analyst*: can use only analytical capabilities of the system, i.e. this user can use the functionality accessible for Information Reader except “Understandability management” and “Understandability assessment”.

In addition, the role system offers the set of fine-grained roles “*Functionality XXX Enabled*” where Functionality XXX is a name of the particular menu item. This role allows the user to access only the particular menu item.

If the user does not have rights to access particular functionality, the corresponding menu items will not be visible as shown on Fig. 47.

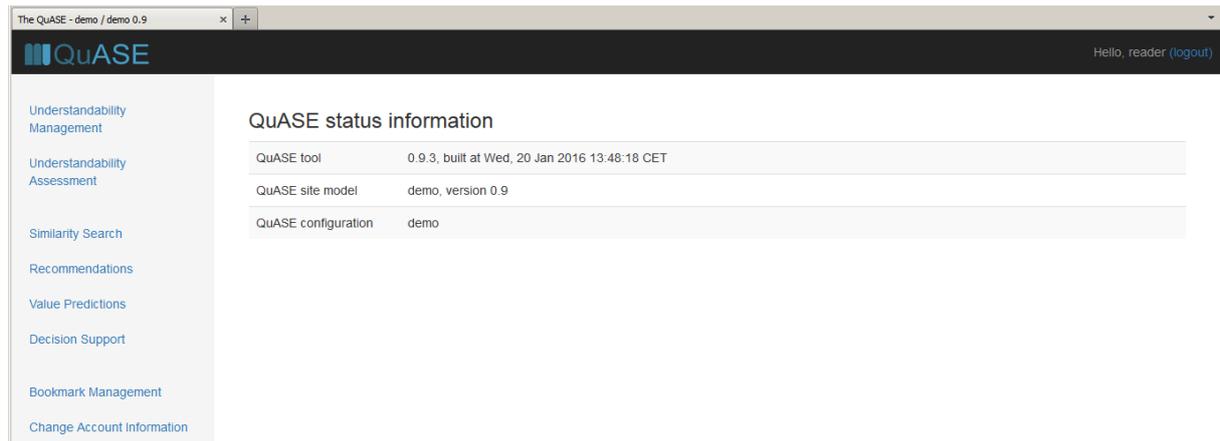


Fig. 47. QuASE tool as visible to the user connected to the Reader role

5.3 Internal endpoint

Internal endpoint (Fig. 48-49) is used to query the knowledge base with SPARQL directly. On Fig. 48, the query is highlighted (note that the necessary SPARQL prefixes are predefined so only the query body has to be specified).

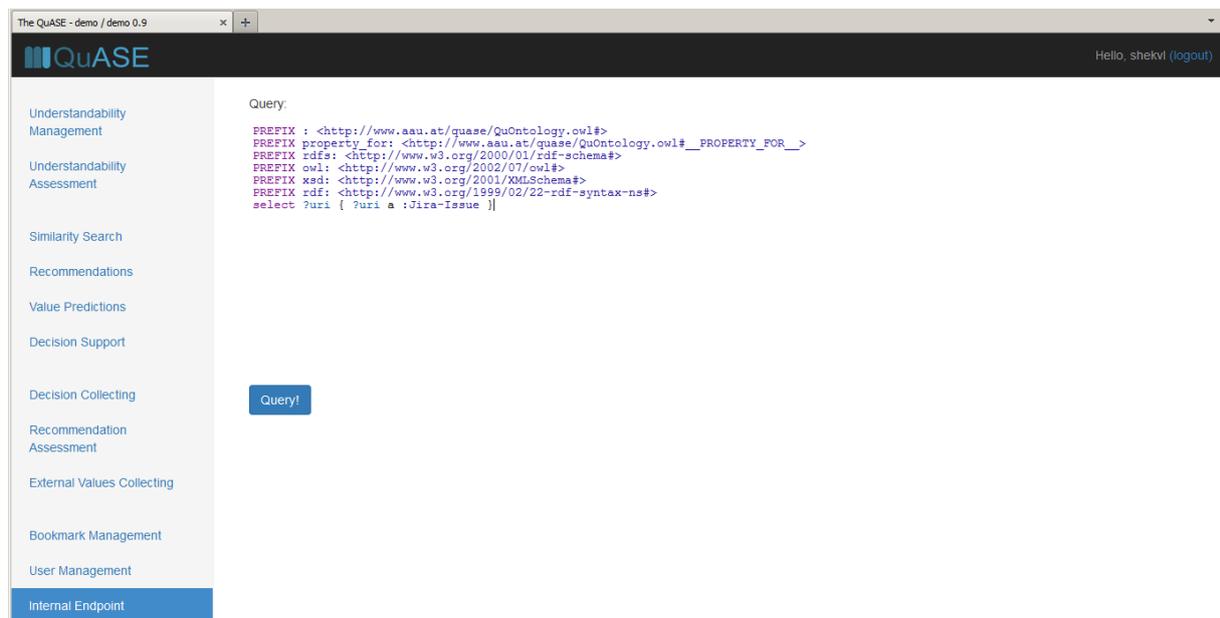


Fig. 48. Internal endpoint: specifying the query

The query results are shown on Fig. 49, if the query results in error, the error message is displayed instead of the results.

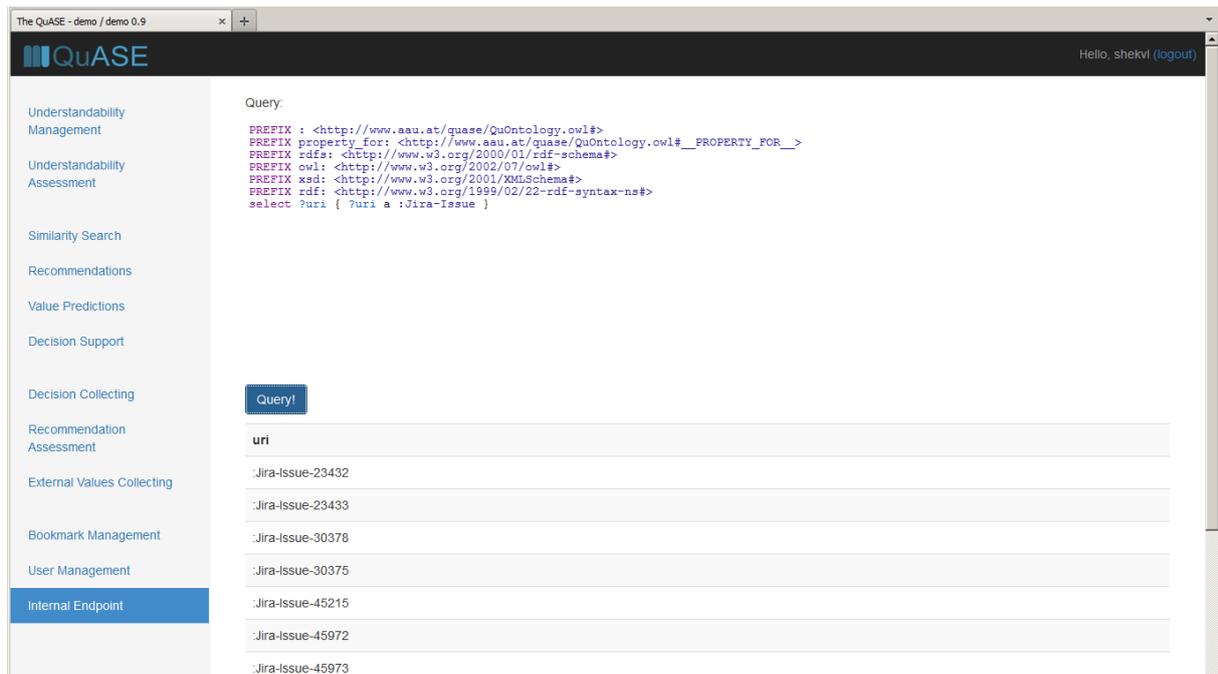


Fig. 49. Internal endpoint: obtaining the results

6 Integrating QuASE and Jira

Another approach to working with QuASE is to use it in embedded mode from the Jira UI with support of the QuASE Jira plugin. Currently, the QuASE support is enabled for issues, users, projects, project versions, and components.

The functionality of QuASE available from Jira is accessible as follows:

1. The element is selected by Jira means;
2. The functionality of QuASE becomes available for this element in an embedded window (screen area) in the Jira UI related to the selected element, the functionality works as in standalone mode, except that the element selected in Jira is pre-selected in QuASE;
3. The available functionality depends on the configuration defined in the site model, but in general, the analysis functionality is available for context units (users, projects, project versions and components), for content units (issues) the understandability improvement functionality is available as well;
4. The understandability assessment functionality is not available because it involves selecting a set of elements through QuASE which is not compatible with the embedded mode where a single element is selected through Jira; the auxiliary functionality (bookmark support, user management, and internal endpoint) is not available as well;
5. On first access of the QuASE functionality, the startup screen (Fig.36) is displayed, which includes besides standard model-related information and the button for synchronizing the knowledge base, the information about the class and the name of the selected document. This name is displayed in green; if the element selected in Jira is not available in QuASE KB (e.g. when the element has been just added to Jira and no KB synchronization has been performed since then), the message about the unavailability of the element in the KB is displayed in red.

6.1 Working with issues

The QuASE functionality is available from the additional "QuASE" tab in the Jira issue screen (Fig. 50).

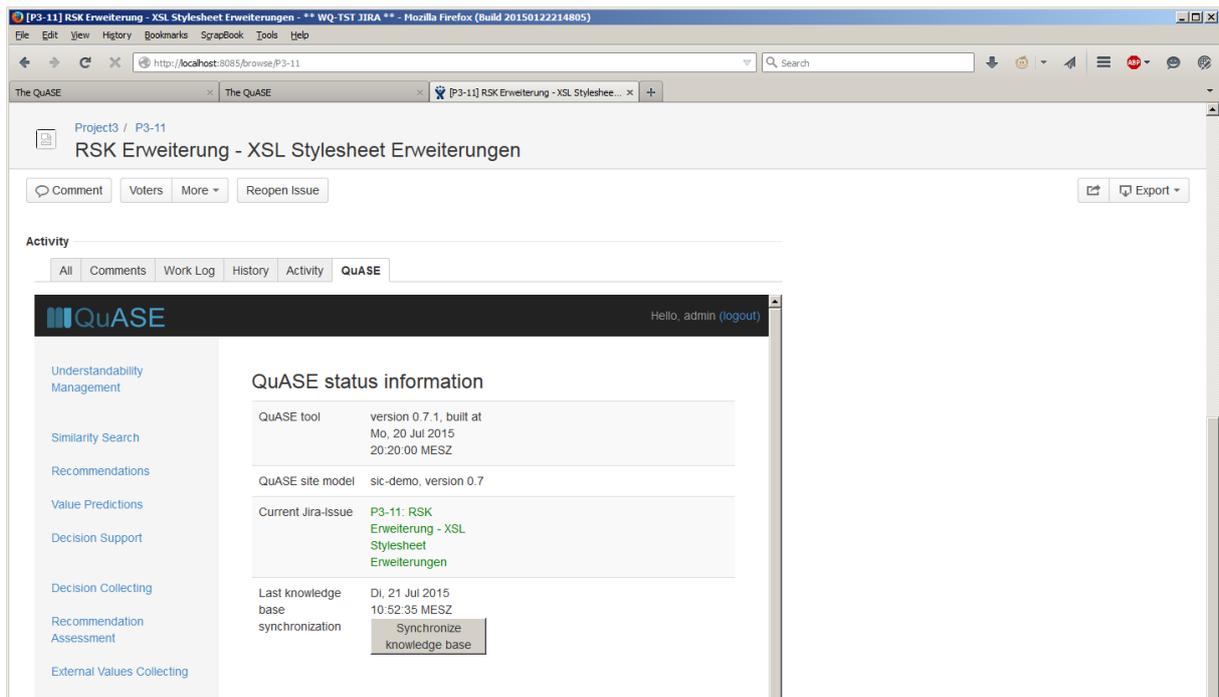


Fig. 50. Jira issue screen with embedded QuASE frame

The user can navigate through the main menu, every activity there will be pre-configured by pre-selecting the element selected in Jira. An example of such pre-configured view of the understandability improvement functionality is shown on Fig. 51: the particular issue is pre-selected in QuASE, all other selectors are available, and so it is possible to provide the understandability improvement of this Jira-selected issue for different contexts and knowledge domains. To select different issue, it is necessary to use Jira navigation functionality.

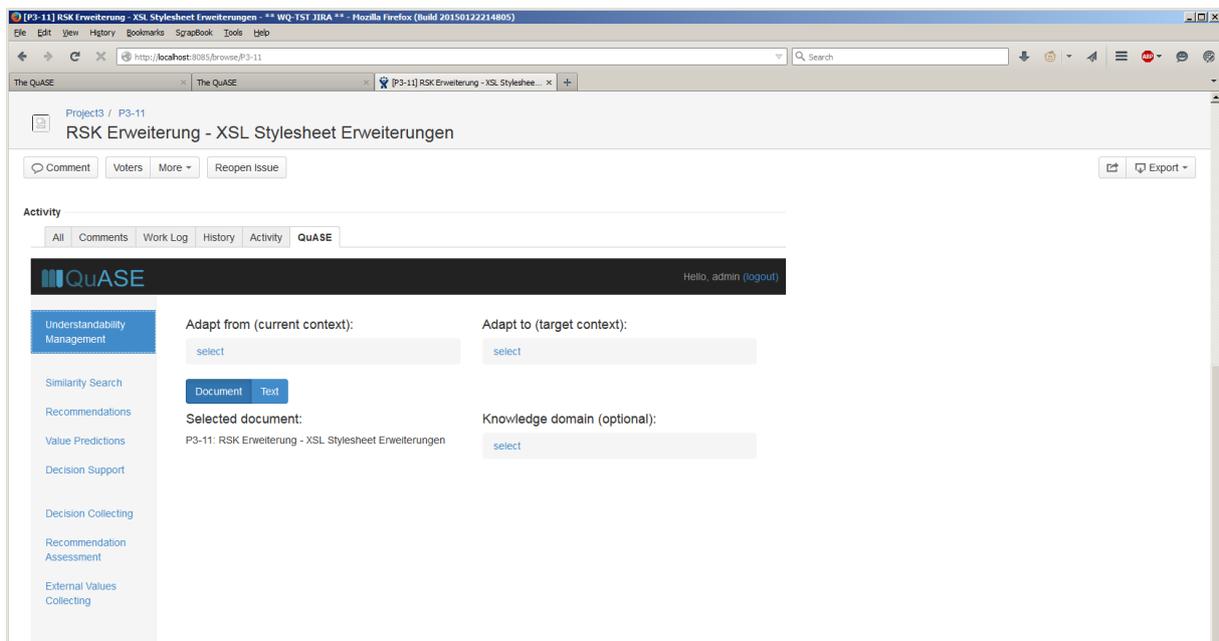


Fig. 51. Understandability improvement startup screen in embedded mode

Additional examples of the QuASE functionality available from Jira for the issues is shown on Fig. 52 and Fig. 53.

Fig. 52 shows decision prediction startup screen with the current issue pre-selected as decision target: the user only needs to select decision kind.

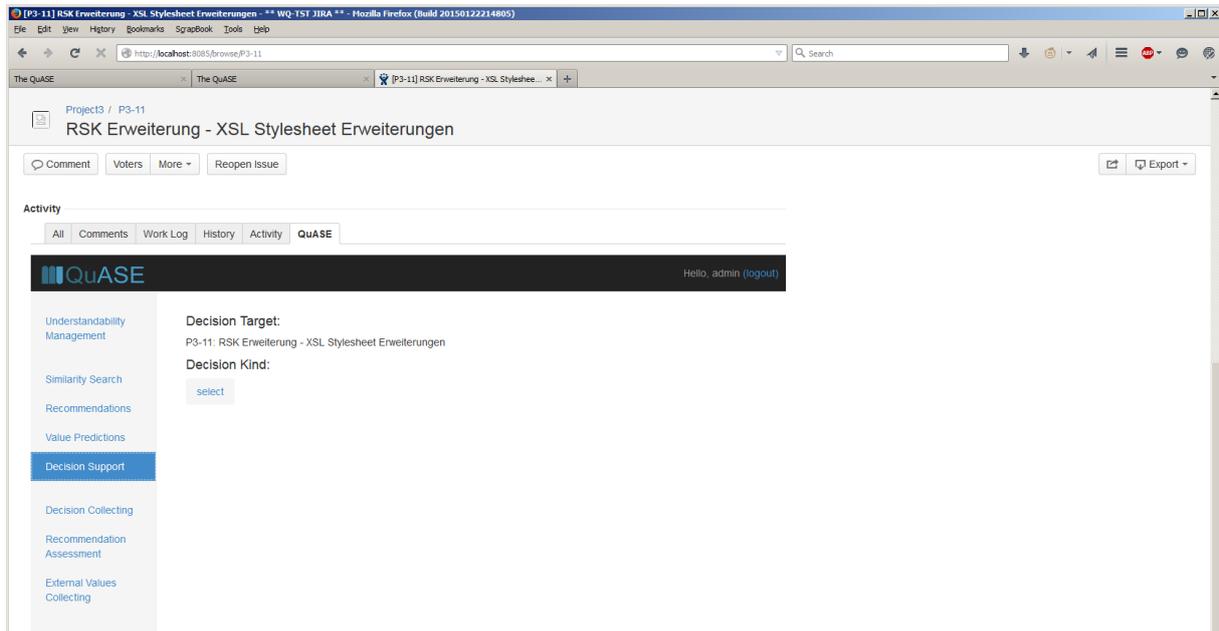


Fig. 52. Decision support startup screen in embedded mode

Fig. 53 illustrates how it is possible to specify values for the external (user-supplied) information from QuASE: again, only attribute values, connections and their properties etc. have to be specified, the current Jira issue is pre-selected. In particular, the management of external units is completely available from Jira embedded mode.

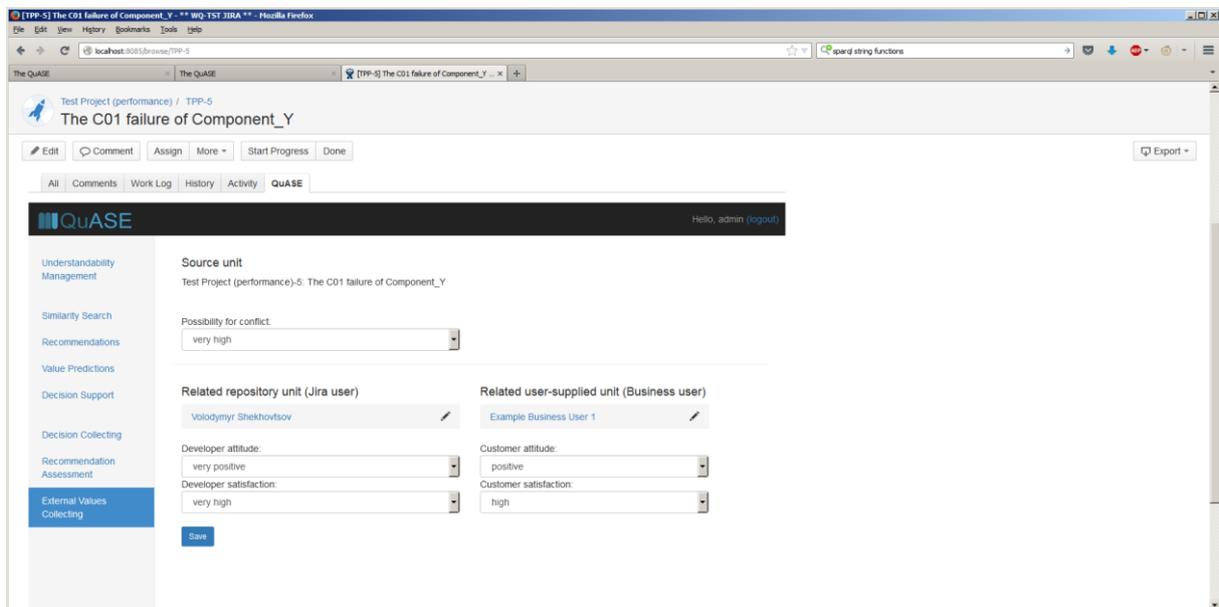


Fig. 53. Specifying external information in embedded mode

6.2 Working with context elements

Fig. 54-57 illustrate accessing QuASE support for context elements. For the users (Fig. 54), the functionality is available from the “User profile” screen and is not delegated to the separate tab. As mentioned before, only understandability management functionality is not available for these units,

all analytical activities can be performed together with specifying the values of the external attributes.

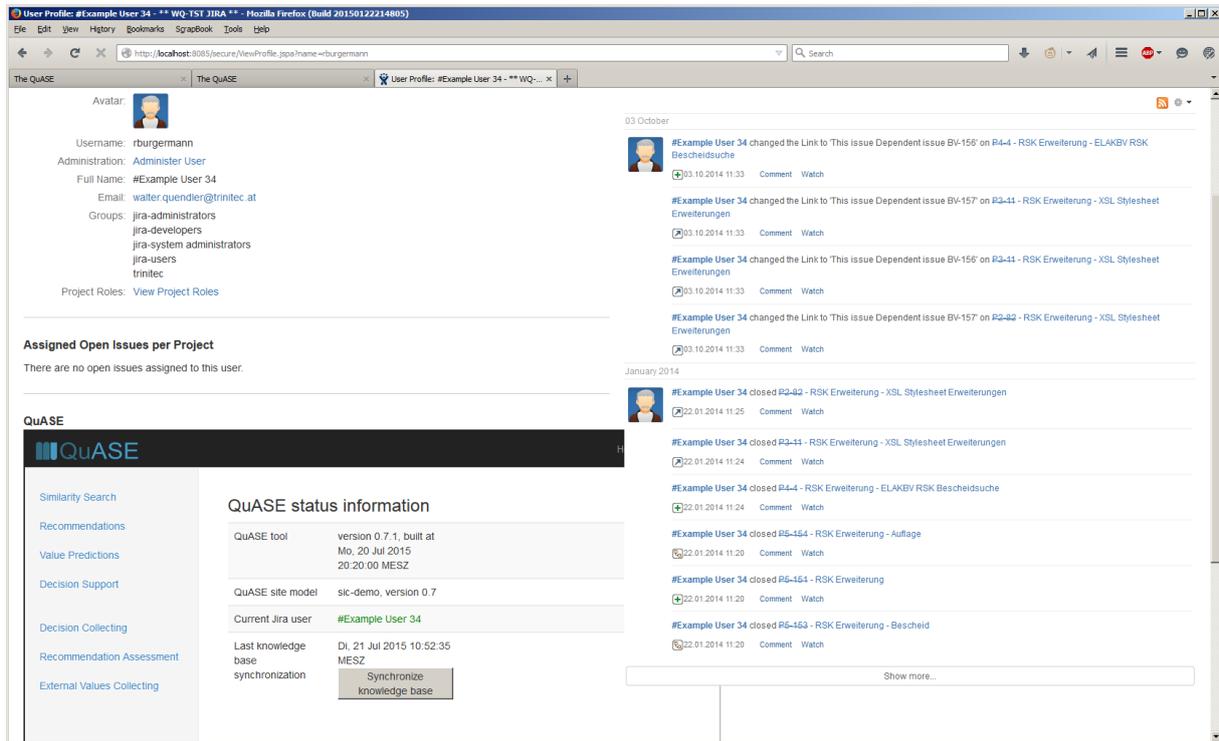


Fig. 54. Jira user profile screen with embedded QuASE frame

For the projects (Fig. 55), project versions (Fig. 56), and components (Fig. 57) the functionality is available in the separate tabs accessible from left-side Jira menu.

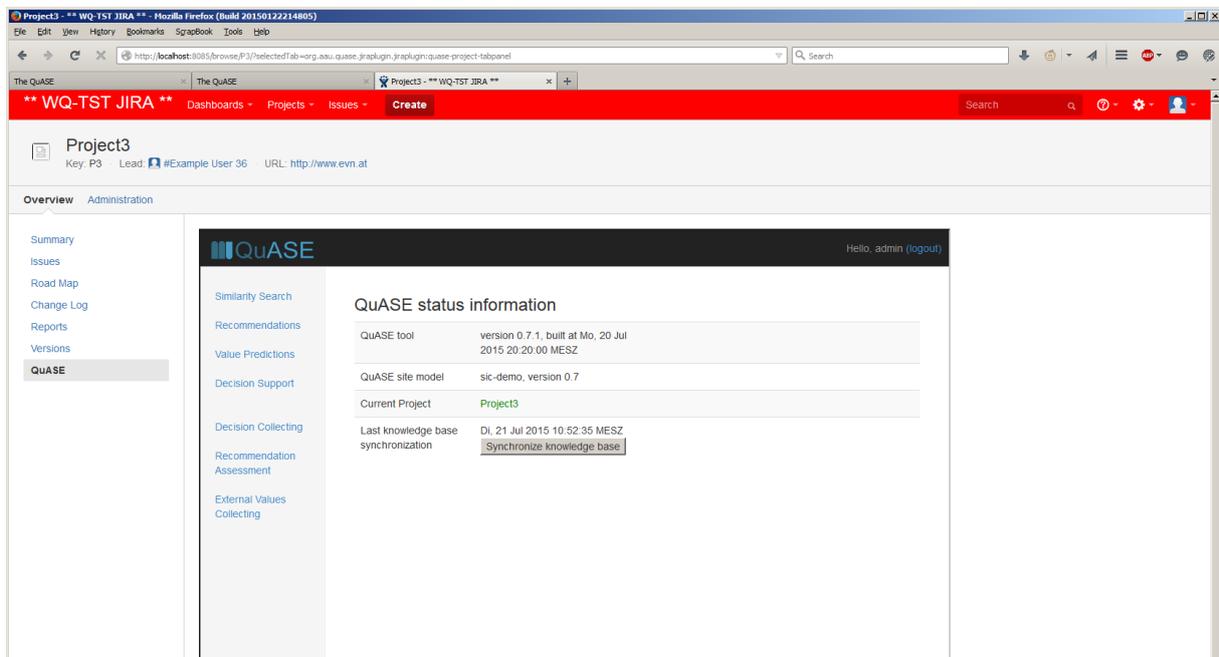


Fig. 55. Jira project screen with embedded QuASE frame

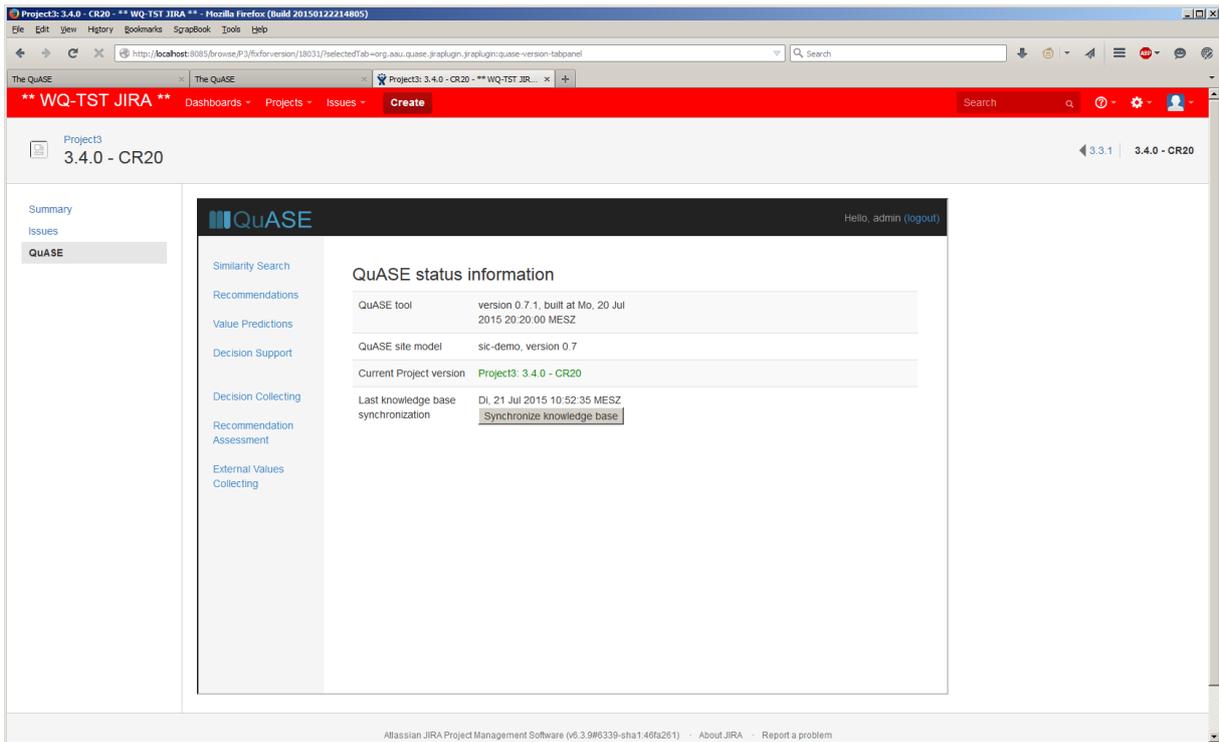


Fig. 56. Jira project version screen with embedded QuASE frame

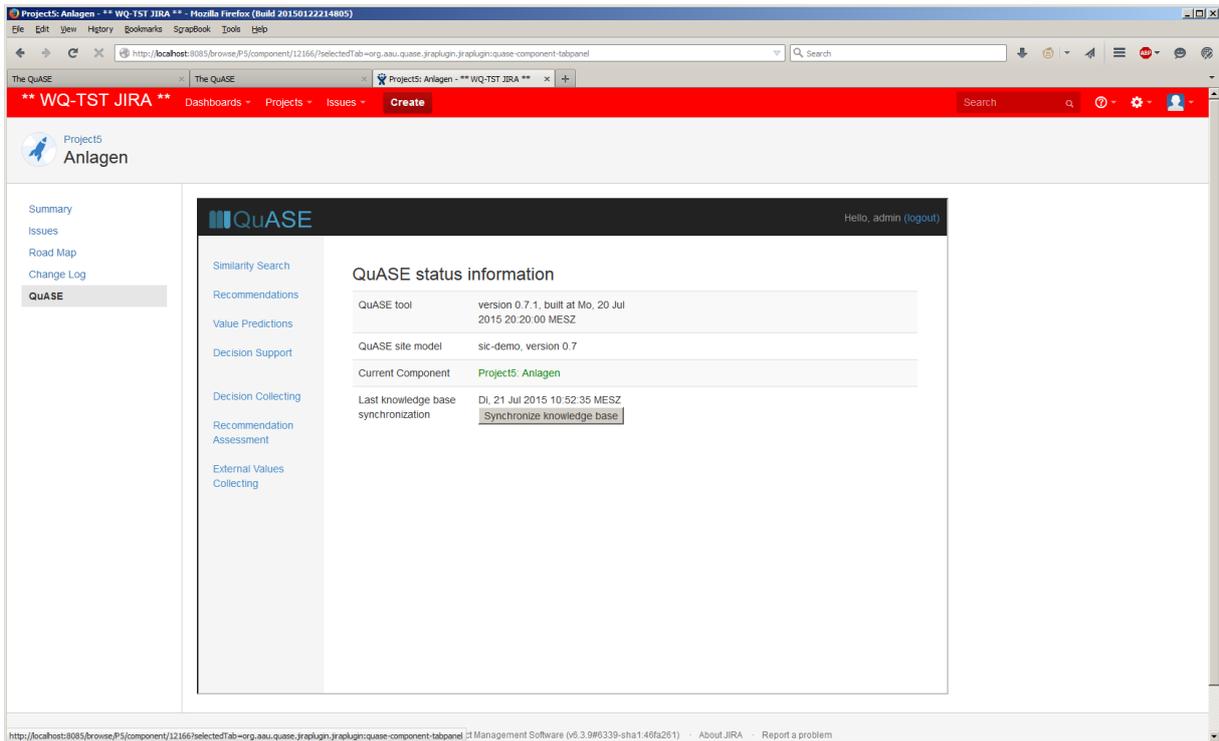


Fig. 57. Jira component screen with embedded QuASE frame