KWIZ, a Knowledge-Acquisition Framework

1 Introduction .................................................................................................................. 2
2 Installation ................................................................................................................... 2
3 Configuration of KWIZ .............................................................................................. 3
   3.1 Knowledge-base Views ..................................................................................... 3
   3.2 Configuration of Views ..................................................................................... 3
4 The KWIZ Knowledge-acquisition Environment ..................................................... 9
   4.1 The Configuration Screen ............................................................................. 10
   4.2 The Navigation Trail / Tree .......................................................................... 11
   4.3 The Tabs ........................................................................................................... 12
   4.4 The Class Region screen .............................................................................. 13
   4.5 The Instance Region screen .......................................................................... 14
   4.6 The Instance Form screen ............................................................................ 15
   4.7 The Instance Creation screen ........................................................................ 16
      4.7.1 How to create new instance ..................................................................... 16
      4.7.2 How to search for instances ..................................................................... 16
   4.8 The Instance Tree Search screen .................................................................... 18
      4.8.1 How to search an instance tree ............................................................. 18
   4.9 The Help screens ............................................................................................ 19
5 Generating HTML Pages for SAGE Guideline Knowledge Bases ...................... 19
6 Appendix: KwizTab Quick Start Notes ................................................................. 21
1 Introduction

KWIZ is built as an extension to Protégé to improve Protégé’s knowledge-acquisition environment. The main components of KWIZ are:

- Customizable high-level views of the knowledge base
- Constrained navigation
- Reuse of existing knowledge bases
- Context-sensitive search and help

The KWIZ knowledge-acquisition environment is built as a tab-plugin of Protégé (Figure 1). In this document, we will first discuss the knowledge-base views supported by KWIZ and how to configure them using the KWIZ knowledge model. Next we will illustrate the main features of the KWIZ environment such as the views, the navigational modes to browse the knowledge base, the library facility and the search features.

The KWIZ package also contains the kb2doc software and configuration files that are needed to generate HTML views of a SAGE guideline knowledge base.

2 Installation

The KWIZ software and the necessary Protégé projects are available as part of the SAGE Guideline Workbench available from http://www.sageproject.net. The
“KwizTab Quick Start Notes.doc” file in the zip archive has installation instructions. It also appears as an appendix of this document. The content of the installation package is configured to create KWIZ workbench for the SAGE guideline project. SAGE members can go directly to Section 4 and 5 for documentation on how to use KWIZ as a SAGE guideline workbench.

3 Configuration of KWIZ

3.1 Knowledge-base Views

KWIZ can provide a domain-specific view of the knowledge model concepts by creating forms that match a clinician’s cognitive model of the domain and that are mapped to components in the knowledge base. These views can be used to hide the complexities of the underlying computable model from the user. The views also allow compartmentalizing the knowledge base so that users are exposed only to parts of the knowledge base that are relevant to them. Views can be used to browse large knowledge bases effectively as they provide quick access to most used parts of the knowledge base.

In KWIZ, a view is a combination of three knowledge base components: a class of instances, a specific instance, and specific slots of an instance. We refer to these components as regions. Thus the ‘class of instances’ component is the Class region and the ‘specific instance’ component is the Instance region.

3.2 Configuration of Views

As part of KWIZ, we have developed a knowledge model that is used to specify the configuration of the knowledge-acquisition environment. Currently, the model is only used to define knowledge-base views. Thus, there are many other concepts such as knowledge-acquisition scripts included in the knowledge model but have not been implemented yet, and hence, the seeming complexity in defining views using the KWIZ model. Figure 2 shows the KWIZ model classes (as defined in the KWIZBase Protégé project). The highlighted classes are the only ones currently implemented and are necessary for specifying the views.
In the following, we will use the terminology where a *domain model* means an ontology in an application domain (e.g., a model of clinical guideline) and a *domain knowledge base* means a knowledge base in that domain (e.g., a hypertension guideline as encoded in the specific guideline model). The major steps in configuring a KWIZ environment for a specific domain are:

1. Copy the KWIZBase project to the directory that contains the domain model. Create an empty Protégé project in that directory, and name the project as, say, “KWIZinDomain.” Include in the KWIZinDomain project the KWIZBase project and the Protégé projects that specify the domain model.

2. Configure views using the KWIZ model and the domain model concepts.

3. Copy the KWIZinDomain project to the directory that contains a specific domain knowledge base project. Note that the same KWIZinDomain project can be used with multiple domain knowledge bases. In our example, if we configure KWIZinDomain to work with a clinical guideline model, that configuration can be used with different knowledge bases, such as the Immunization Guideline or the Diabetes Guideline, that use the same clinical guideline model.

---

**Figure 2 KWIZ knowledge model**

![Diagram of KWIZ knowledge model](image-url)
We will illustrate the configuration of the views using the KWIZ model with an example. Figure 3 shows parts of a clinical guideline model. In our example, we want to create views that will expose only the concepts that have been highlighted. We want to create two views: a Clinical Guideline view and an Expressions view. The Clinical Guideline view will include a specific instance of the Guideline class and all instances of the Guideline_Resources class. Further, we want to expose only the highlighted slots (see Figure 3) of the Guideline class. The Expressions view will refer to all instances of Criterion, Variable and Function classes.

Each view is specified as an instance of the Wizard class of the KWIZ model (Figure 2). Thus, in our example, we will create two instances: Clinical Guideline and Expressions. In the KWIZ knowledge-acquisition environment, these two instances will be presented as options in the configuration screen (Figure 12). The main slot of the Wizard class is:

- **Region Script** – defines the regions of the view.

Figure 4 shows the regions of the Clinical Guideline view. It has two regions: Guideline_Resources (a Class region which specifies the class whose instances will be part of the configured KWIZ knowledge-acquisition environment for the domain) and Guideline (an Instance region which displays the content of

![Diagram of a clinical guideline model and the Guideline class](image)

**Figure 3** A clinical guideline model and the Guideline class
a particular instance). In the KWIZ environment, each region appear as tabs (Figure 1), and the order of the regions in the graph as seen in Figure 4 dictates the order of the tabs.
Figure 5 shows the Protégé form of a Class region. The main slot to be filled is: **Domain Class** - the domain class that is associated with the Class region. In our example, the clinical guideline domain class **Guideline_Resources** is associated with the Class region, Guideline_Resources (Figure 5). In the KWIZ environment, the tab *Guideline Resources* displays all instances of the domain class **Guideline_Resources** (Figure 11).

![Figure 5 Guideline Resources, a Class region](image)

Figure 6 shows the Protégé form of an Instance region. The main slots are: **Domain Class** – the domain class associated with the Instance region **Component Script** – the subset of slots of the domain class that should be shown with the instance.

![Figure 6 Guideline, an Instance region](image)

As mentioned before, with an Instance region, we can choose to expose only a subset of slots of the associated domain instance. Figure 7 shows the Component Script for the Guideline region. This specification of the Guideline Instance region manifests in the KWIZ environment as the tab Guideline. The tab allows the user to view a specific instance of the domain class. The slots as specified in the Component Script appear as sub-tabs (Figure 16). As mentioned earlier, the KWIZ model allows the specification of complex knowledge-acquisition process that has not been implemented yet in the KWIZ GUI environment. Hence, specifying the components (slots of the domain instance)
Draft

involves more steps than what may seem necessary. The two values necessary for specifying a component are the domain class and the appropriate slot of the domain class. In our example, for the Recommendation Set component, the domain class is **Guideline** and the domain slot is **recommendation_set**, both of which are part of the Clinical Guideline model. Figure 8 shows the **Recommendation Set** component that has a subscript. The subscript (Figure 9) has one task, a **Domain Instance**, in the task sequence. The Domain Instance task (Figure 10) has the slots **domain slot name** and **domain parent class** filled with the appropriate values.

Thus, the steps to configure views are: define the views, and, for each view, define the regions (Class region and/or Instance region) that make up the view, and then, for each Instance region define the domain slots that need to be exposed in the KWIZ environment.
4 The KWIZ Knowledge-acquisition Environment

KWIZ is a Protégé tab plugin. Once the KWIZ files have been appropriately installed, the user will be able to configure the Protégé environment to display the KWIZ tab as shown in Figure 11.

Figure 11 KWIZ knowledge-acquisition environment
4.1 The Configuration Screen
The Configuration screen (Figure 12) allows the user to select a knowledge base view, and also to specify any previously developed knowledge bases as part of a library. KWIZ will load these knowledge bases in the background and the user will be able to view instances in the library and copy selected instances into the current knowledge base.

Figure 12 KWIZ Configuration Screen
4.2 The Navigation Trail / Tree

As and when the user navigates to different parts of the knowledge base via the KWIZ screens, the Trail tab (Figure 13) on the navigation panel on the left-hand side of the KWIZ tab shows the path taken by the user. By double-clicking on any entry on the path, the user can back-track to that point. The Tree tab on the navigation panel displays the instance hierarchy. The user can double-click on a node (instance) on the tree to view that instance.
4.3 The Tabs

The regions of the knowledge base views are presented as subtabs of the KWIZ tab (Figure 14). With an Instance region tab, the sub-tabs display the slots of the domain instance. The configuration of the views, regions and slots is explained in the Configuration of Views section.

Figure 14 Components of a view presented as tabs
4.4 The Class Region screen

The Class Region screen displays all the instances of a domain class (Figure 15). This same UI is also used, when traversing the instance hierarchy, to display the values of a multiple-cardinality slot of instance value type (see Figure 16 that displays values of the recommendation_set slot of the Guideline instance). The screen has two main sections.

The top section displays a list of buttons that correspond to a set of tasks that the user can perform.

The main section displays a list of instances, and a preview area. The user can select any entry on the list to display the corresponding instance in the preview area.

The user can perform the following tasks:

- **View** Select the instance on the list. Click on the View button. A shortcut to this task is to double-click the instance on the list.
- **Create** An instance can be added to the list by clicking on the Create button. This will take the user to another screen where the user can either create a new instance, copy one from the library, clone or refer to an already created instance in the current knowledge base.
- **Delete** An instance can be deleted from the list by selecting the instance and clicking on the Delete button.
- **References** Knowledge base elements (classes and instances) that refer to a selected instance can be displayed by clicking on the References button.

Figure 15 A Class Region screen; also used to display a list of instances
4.5 The Instance Region screen

This screen allows the user to select a specific instance. It displays the configured slots of the instance as sub-tabs (Figure 16).

Figure 16 An Instance Region screen
4.6 The Instance Form screen

The Instance Form screen (Figure 17) displays a Protégé instance. The screen has two main sections. The top section displays the name of the instance that you are browsing and a list of buttons that correspond to a set of tasks that can be performed. The main section displays the instance form that uses some special slot widget buttons.

The buttons in the top section are:

- **Search** When the user clicks this button a search form is displayed that allows the user to search the instance tree with the current instance as the root (i.e., all instances that are referenced directly or directly from the root instance).
- **References** Knowledge base elements (frames) that refer to an instance can be displayed by selecting that instance and by clicking on the References button.

The special slot widget buttons are:

- **Allows the user to zoom-in to a slot to display the form that lists the slot values.**
- **Allows the user to zoom-in to a slot of single cardinality to display the form that shows the slot value. The button can also be used to zoom-into a specific instance in the case of multiple cardinality slots.**
- **Allows the user to add a value to the slot. The value can either be a new instance or an instance from the library or local knowledge bases.**

Figure 17 An Instance Form screen
4.7 The Instance Creation screen

The Instance Creation screen (Figure 18) allows you to create an instance of a class. There are two situations where one would want to create an instance: 1) to add to the list of instances of a class (Figure 15), and 2) to add a value to a slot of type Instance (Figure 16). This screen has two main sections. The top create/search section displays a create interface that allows you to create a new instance of an allowed class type. It also displays a search interface that allows you to search for instances in the library or in the local (working) knowledge base.

The main search results section displays the results of the search and buttons corresponding to tasks that can be performed.

4.7.1 How to create new instance

The create/search section allows you to create new instances. The first step is to specify the type of instance you want to create. When creating an instance to add to a list of instances of a class (situation 1), the allowed type will be the Class itself, and, if the Class is abstract, any of its concrete sub-classes. When creating an instance to be a value of a slot (situation 2), the allowed type will be the allowed types of the slot. If there is only one allowed type, then that type will already be populated in the Value Type(s) field. If there is more than one allowed type, click on the Select button to display the allowed types. Select the type from the list. The Value Type(s) field will be populated with that selection. Now click on the Create button. An instance of the specified type will be created and added to the list of instances of the class (situation 1) or as a value of a slot (situation 2). The new instance is then displayed for edition.

4.7.2 How to search for instances

The first step is to specify the search criteria in the top create/search section. There are two parts to the search criteria: 1), the types of instances you want to search and 2) the search terms to narrow the search further. If there is only one allowed type, then that type will already be populated in the Value Type(s) field. If there is more than one allowed type, click on the Select button to display the allowed types. Select the types of instances you want to search on from the list. The Value Type(s) field will be populated with that selection. You can further narrow the search by entering terms in the Add terms to narrow search text field. You can specify * as a wild card. These terms will be matched against the browser text of the instances. Click on the Search button to search the library and local knowledge bases.

The main search results section displays instances in the library and local knowledge base that match the search criteria. There are several tasks that the user can perform in the search results section:

Viewing an instance Select an instance on the list to display the instance form on the right. The instance can also be viewed as a pop-up by
selecting the instance on the list and clicking on the View buttons, or by just double-clicking on the instance.

**Copying an instance from the library** Select the instance from the library list. Click on the Copy button. The entire instance tree starting from the selected instance as the root is copied over to the local knowledge base. The copy function does not copy instances in the source instance tree for which there are instances that have identical slot values in the local knowledge base.

**Referring to an instance in the local knowledge base:** Select the instance from the local list. Click on the Refer-to button.

**Cloning an instance from the library** Select the instance from the local list. Click on the Clone button. The selected instance is then cloned i.e. the instance is copied up to one level.

Note that the instances created by any of these tasks are added to the list of instances of the class (situation 1) or as a value of a slot (situation 2). Also, the instances are displayed for any further editing after the task is performed.

**Figure 18 Create Instance screen**
4.8 The Instance Tree Search screen

The search screen (Figure 19) has two main sections. The top search criteria section displays a search interface that allows you to search an instance tree for specific knowledge elements with the current instance as the root of the tree. The main search results section displays the results of the search.

4.8.1 How to search an instance tree

The first step is to specify the search criteria in the top search criteria section. There are two parts to the search criteria: 1), the types (classes) of instances you want to search and 2) the search terms to narrow the search further. Click on the Select button to display the types. Select the types of instances you want to search on from the list. The Value Type(s) field will be populated with that selection. You can further narrow the search by entering terms in the Add terms to narrow search text field. You can specify * as a wild card. These terms will be matched against the browser text of the instances on the instance tree. Click on the Search button to search the instance tree.

The main search results section displays instances in the instance tree that match the search criteria. The user can select an instance on the list and click on the References button to see what knowledge elements refer to the selected instance. The user can double-click on an instance to zoom-in to that instance (i.e. view that instance in the instance form).

Figure 19 Search the Instance Tree screen
4.9 The Help screens

Most of the KWIZ screens have a Help button. Users can access two types of help (Figure 20) by clicking on the Help button: (1) domain-specific information on the screen contents, and (2) KWIZ-specific information on what the screen is all about and the tasks that a user can perform on that screen.

![Figure 20 Help screens](image)

5 Generating HTML Pages for SAGE Guideline Knowledge Bases

The capability to generating HTML pages for SAGE guideline knowledge bases is done through the Kb2Doc tab in Protégé. The software and necessary Protégé projects are part of the KWIZ package, which is available as part of the SAGE Guideline Workbench (http://www.sageproject.net). The “KwizTab Quick Start Notes” appendix of this document has installation instructions.

Start Protégé with the directory `edu.stanford.smi.protegex.kwiz` in your Protégé plugins directory. If Kb2Doc does not show up as one of the tabs, go to the Project/Configure menu item and check `Kb2DocTabWidget` in the list of available tab plugins.

Figure 21 shows the initial screen of the Kb2Doc tab. There are four configuration parameters:

1. *Doc Templates* is a Protégé project that specifies how the SAGE guideline ontology should be translated into XML. It should be `kb2doc/sagekbdescription.pprj` for the current KWIZ installation.

2. *Doc Style Sheet* is a file containing XSLT to translate XML to HTML. It should be `kb2doc/sagekb2doc.xslt` in the current KWIZ installation.
3. **Output File Name** is the base component of the XML and HTML files that will be generated (e.g., SAGECAPGuideline.xml and SAGECAPGuideline.html)

4. **Directory** is the path to the directory to which Kb2Doc will store the generated files.

Once the four parameters are specified, click the **Generate Doc View** button. Figure 22 shows the XML output for the SAGE CAP guideline. You can toggle between XML and HTML views by selecting the **XML View** and **HTML View** tabs. You can also use an external browser to view the XML and HTML output by clicking the **View in Browser** button.

If Kb2Doc tab fails to generate the proper XML and HTML files, the most likely cause is that kb2doc/sagekbdescription.pprj has not kept up with changes in the guideline model.

![Figure 21 Initial screen of the Kb2Doc tab](image)
6 Appendix: KwizTab Quick Start Notes

1. Copy the directory edu.stanford.smi.protegex.kwiz to Protege plugins directory.
2. Copy the directory edu.stanford.smi.protegex.instance_tree to Protégé plugins directory.
3. Copy the kwiz project files (KWIZ and KWIZBase pins, .ponts and .pprj files) in SAGEkwizKb directory to *your guideline project directory*.
4. Copy the kb2doc directory to any location, preferably at the level of your other guideline projects.
5. Open your guideline project (e.g. SAGE_CAP.pprj)
6. In the top Protégé menu, go to Project / Configure. Select KwizTabWidget.
7. You should see the KWIZ tab and KB2DOC tab
8. In the KWIZ tab, select (from a drop-down list) the view that you want (currently only one view is available) and add the library projects that from which you want to copy instances.

9. You'll see tabs like Variable, Function, Criterion, Recommendation_Set that display all instances of these classes, and a Guideline tab for which you have to “Open” an instance of Guideline to traverse the content of that instance.


11. Now you can click on any other tab and start navigating.