CS3733-D01: Software Engineering

The KnowledgeBase Project : Subsystem Interfaces

This document outlines the initial subsystem interfaces that we will use for the KnowledgeBase project. This document is subject to revision as teams find revisions and updates needed to implement the project. We will freeze revisions a few days before the project due date.

This document contains three main sections: a description of the classes that cross subsystem interfaces, signature definitions, and the import and export signatures for each subsystem in the project.

1 The Classes

1.1 Defined in the Server Component

The server-id is a value chosen by the server to help locate the data structure in the knowledge base to which an item corresponds. Each server group can decide what to use as this value.

1.1.1 ItemData%

Attributes:

- text : string (initialization)
- id : server-id (initialization)

Methods:

- get-text : $\rightarrow$ string
- get-id : $\rightarrow$ server-id

1.1.2 HelpItem%

Attributes:

- notes : set-of-string
- fixes : set-of string

Methods:

- add-note : string $\rightarrow$ void
- add-fix : string $\rightarrow$ void
- for-each-note : (note $\rightarrow$ void) $\rightarrow$ void
  Applies the given function to every note in the notes set.
- for-each-fix : (fix $\rightarrow$ void) $\rightarrow$ void
  Applies the given function to every bug-fix in the notes set.

1.1.3 Instruction% (extends HelpItem%)

Attributes:

- data : ItemData (initialization)

Methods:

- get-data : $\rightarrow$ ItemData

1.1.4 Solution% (extends HelpItem%)

Attributes:

- data : ItemData (initialization)

Methods:

- get-data : → ItemData

1.1.5 Question% (extends HelpItem%)

Attributes:

- text : string (initialization)
- poss-answers : set-of-ItemData (initialization)

Methods:

- get-text : → string
- for-each-answer : (ItemData → void) → void Applies the given function to each itemdata corresponding to the possible answers for a question

1.2 Defined in the KnowledgeBase Component

1.2.1 KnowledgeBase%

Attributes:

- topics : list-of-Topic

Methods:

- add-topic : string KnowledgeBase → topic

  Adds a topic to the KnowledgeBase with the given string as a description and returns the topic.

- search : (Topic → boolean) (SubTopic → boolean) (Node → boolean) → list-of-value

  Search visits every topic, subtopic, and node in the knowledgebase and creates a list of all topic, subtopic, and node objects that match the search criteria. The three functions given as input specify the search criteria. Each function is a predicate that indicates whether the given item matches the criteria. The first function is applied to each Topic, the second to each SubTopic, and the third to each Node. The output list consists of all Topic, SubTopic, and Node objects for which the corresponding function evaluated to true.

1.2.2 Topic%

Attributes:

- description : string (initialization)
- contents
- subtopics : list of SubTopic

Methods:

- add-subtopic : topic string contents KnowledgeBase → subtopic

  Adds a subtopic to the given topic in the KnowledgeBase. The subtopic has the description given in the string argument and a (new) node with the given contents. Returns the subtopic.

- get-subtopics : → list-of-SubTopic

1.2.3 SubTopic%

Attributes:

- description : string (initialization)
- node : Node

Methods:

- get-description : → string
- get-node : → node

1.2.4 Node%

Attributes

- contents : value (initialization)
- next : list-of-node

Methods:

- get-contents : → value
- set-contents : value → void

  Sets the contents of the given node to the given value (the new contents), replacing the old contents.
- insert-node : contents → node

  Create a new node with the given contents and the next-nodes of the current node. Sets the next-nodes of the current node to the new node. Returns the new node.
- insert-next-node : contents → node

  Create a new node with the given contents and no next nodes. Insert the new node into the list of next nodes for the current node. Returns the new node.
- for-each-node : (node → void) → void

  Given a function on nodes, visit every node reachable from the current node and apply the function to that node. Does not return anything.

2 The Signatures

2.1 Interfaceˆ

- make-symbol-datum-req : string → symbol-datum-req

  Makes a value to request a symbol with the given string as a description of the datum.
- make-string-datum-req : string → string-datum-req

  Makes a value to request a string with the given string as a description of the datum.
- make-number-datum-req : string → number-datum-req

  Makes a value to request a number with the given string as a description of the datum.
- make-select-one-datum-req : string list-of-option → select-one-datum-req

  Makes a value to request that exactly one option from the list of options be chosen. The string describes the set of choices. The functions in the options should consume no arguments.

- make-option : string func → option

  Makes an option element with the given string as a description of the option and the given function as the callback for the option. Menu option and select-one-choice callbacks should be thunks (functions taking no arguments) and form option callbacks should take the number of arguments needed to create the data form.

- make-form : string list-of-datareqs list-of-options → form

  Creates a form value containing the given data requests and options with the given string as a description of the form.

- make-menu : string list-of-options → menu

  Creates a menu value containing the given options with the given string as a description of the menu.

- make-download : string url → download

  Creates a download value containing two strings, the first a description of the download and the second the URL for the download.

- make-instructions : list-of-string-or-download list-of-options → instructions

  Creates an instructions value containing a list of strings and downloads (the steps corresponding to the instructions) and a list of options.

- display-menu : menu → void

  Displays the given menu.

- display-form : form → void

  Displays the given form (and its options).

- display-msg : string → void

  Displays the given string as a message.

- display-instructions : instructions → void

  Displays the given instructions (and its options).

- display-download : download → void

  Displays the given download

2.2 ControllerSystem^

- startTech : → void

  Starts a version of the client software with tech privileges

- startCustomer : → void

  Starts a version of the client software without tech privileges

2.3 ControllerClasses^

- ItemDataExt% : an class extension intended for ItemData% that accepts the same initialization arguments.
- HelpItemExt% : an class extension intended for HelpItem% that accepts the same initialization arguments.
- InstructionExt% : an class extension intended for Instruction% that accepts the same initialization arguments.
- SolutionExt% : an class extension intended for Solution% that accepts the same initialization arguments.
- QuestionExt% : an class extension intended for Question% that accepts the same initialization arguments.

2.4 Controller^

The combination of ControllerSystem^ and ControllerClasses^.

2.5 ServerSystem^

- start-server → void

  Starts the server. This is only needed when using networking, since otherwise starting the controller code would start the whole system.

2.6 ServerClasses^

- ItemData%, HelpItem%, Instruction%, Solution%, Question%

2.7 ServerOperations^

- get-product-list : → list-of-ItemData

  Returns a list of ItemData, each containing the name of a product and the server's id info for that product.

- get-problem-list/product : product-id → list-of-ItemData

  Returns a list of ItemData, each containing the description of a problem for the product with the given id info and the server's id info for that product.

- add-product : string → void

  Adds a product with the given string as a name to the knowledge base.

- add-problem/product : string server-id → void

  Adds a problem with the given string as a description to the product with the given server-id in the knowledge base.

- add-note : server-id string → void

  Adds the given string as a note annotation to the help-item with the given server-id.

- add-download : server-id download → void

  Adds the given download to the help-item with the given server-id.

- get-next-help-item : server-id → help-item

  Returns the help-item that follows the help-item with the given server-id

- insert-instruction : server-id string → server-id

  Inserts an instruction with the given string as text after the help-item with the given server-id. Returns the server-id of the new instruction.

- insert-question : server-id string list-of-string → server-id

  Inserts a question with the single string as the question text and possible answers as given in the list of strings. The new question is inserted after the help-item with the given server-id. Returns the server-id of the new question.

- insert-answer : server-id string → server-id

  Inserts a new possible answer for the question with the given server-id. The given string provides the text for the possible answer. Returns the server-id of the question to which the answer was added.

- insert-solution : server-id string → server-id

  Inserts a new solution after the help-item with the given server-id. The given string provides the text of the solution. Returns the server-id of the new solution.

- modify-help-item : server-id new-text → void

  Modifies the text of the help-item with the given server-id to be the new-contents in the knowledge base.

- add-bug-fix-ref : server-id string → void

  Adds the given string to the bug-fix-references for every help-item in the knowledge base accessible from the help-item with the given server-id.

- search-kb : list-of-symbol → list-of-ItemData

  Searches the knowledge base for products, problems, and help-items that contain all of the keywords in the given list. Returns a list of ItemData for those items that match the keywords.

2.8 Server^

A combination of ServerClasses^ and ServerOperations^.

2.9 KnowledgeBase^

- create-kb : → KnowledgeBase

  Returns a new KnowledgeBase.

2.9.1 Common^

Common contains functions that may be useful to several subsystems, as well as functions for raising and recognizing exceptions.

- sort : (elt elt → boolean) list-of-elt

  Take a function that compares two elements and returns true if the first is "less than" the second. Returns a list of the elements in the input list, sorted in increasing order according to the function.

- raise-help-item-locked-exn : → void
- help-item-locked-exn? : value → boolean
- raise-no-matches-found-exn : → void
- no-matches-found-exn? : value → boolean
- raise-product-name-exists-exn : → void
- product-name-exists-exn? : value → boolean

3 The Subsystem Imports and Exports

ControllerSystem^ and ServerSystem^ are the exports of the entire product. Note that the Network components may export additional functions to each other; those are up to the network teams, who may add those exports as needed.

3.1 Interface Component

Imports: Common^
Exports: Interface^

3.2 Controller Component

Imports: Interfaceˆ Serverˆ Commonˆ
Exports: Controllerˆ

3.3 Client Network Component

Imports: ServerClassesˆ Commonˆ
Exports: Serverˆ

3.4 Server Network Component

Imports: Serverˆ Commonˆ
Exports: ServerSystemˆ

3.5 Server Component

Imports: KnowledgeBaseˆ ControllerClassesˆ Commonˆ
Exports: Serverˆ

3.6 KnowledgeBase Component

Imports: Commonˆ
Exports: KnowledgeBaseˆ