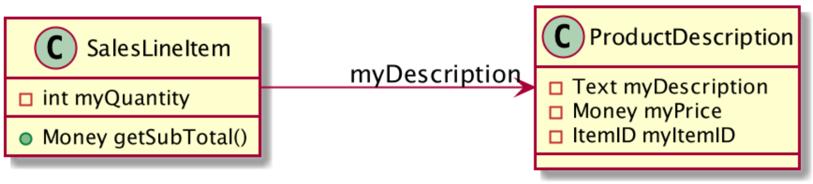
Mapping Design to Code

Mikael Svahnberg*

2016-04-21

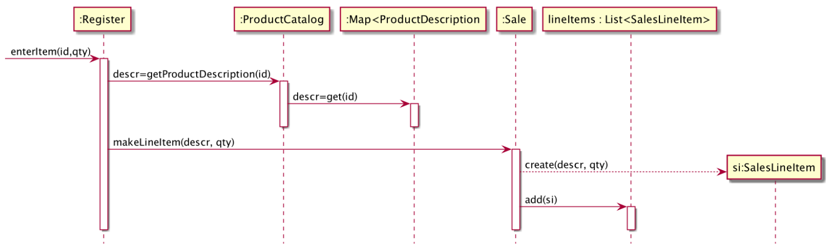
1 Warmup

1.1 Example: From Class Diagram to Code EXAMPLE



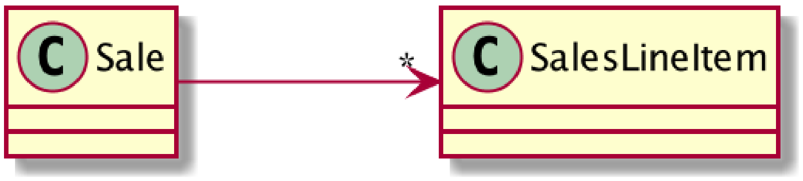
```
public class SalesLineItem {
private int myQuantity;
private ProductDescription myDescription;

public SalesLineItem(ProductDescription theDescription, int theQuantity) {...};

public Money getSubTotal() {...};
}
```

1.2 Example: From Interaction Diagrams to CodeEXAMPLE



*Mikael.Svahnberg@bth.se

1.3 Example: Collections — Example



```
public class Sale {
  private List<SalesLineItem> myItems = new ArrayList<SalesLineItem>;
}

class Sale {
private:
  std::list<SalesLineItem*> myItems;
}
```

1.4 Discuss: Order of Implementation — Discussion

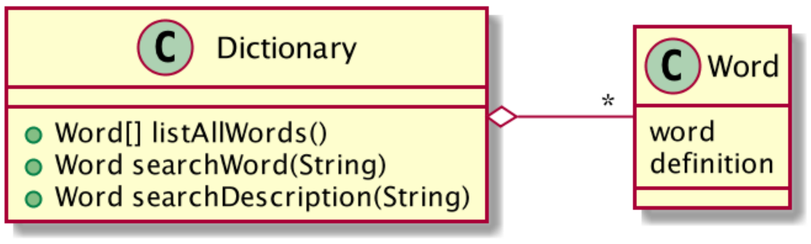
- In which order should classes be implemented?
  - Larman: "Least coupled to most coupled"
  - Other suggestions:
    - Use case per use case, create stubs first, fill them out as you go.
    - First write test cases per use case, then add methods to classes (and create classes) to pass the tests.
    - First write interfaces for all classes, then inherit and implement the classes

2 Dictionary Example

2.1 Task

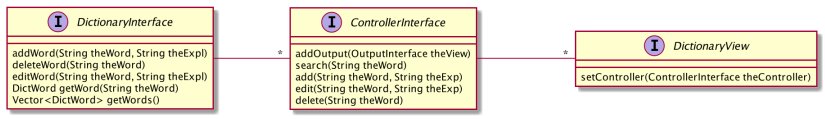
1. Dictionary Write a dictionary program where you have words and their definitions.
   - Users shall be able to browse all words.
   - Users shall be able to search for words
   - Users shall be able to search for definitions.
   - The system shall maintain a log of activities.
   - Other requirements:
     - The system shall use a graphical user interface
     - The system shall store the words and their definitions between sessions.

2.2 Conceptual Model



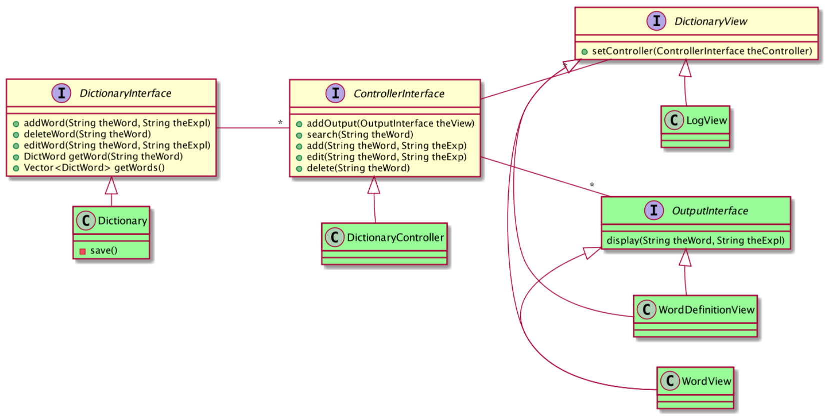
2.3 Class Diagram I

Basic Structure (Interfaces)

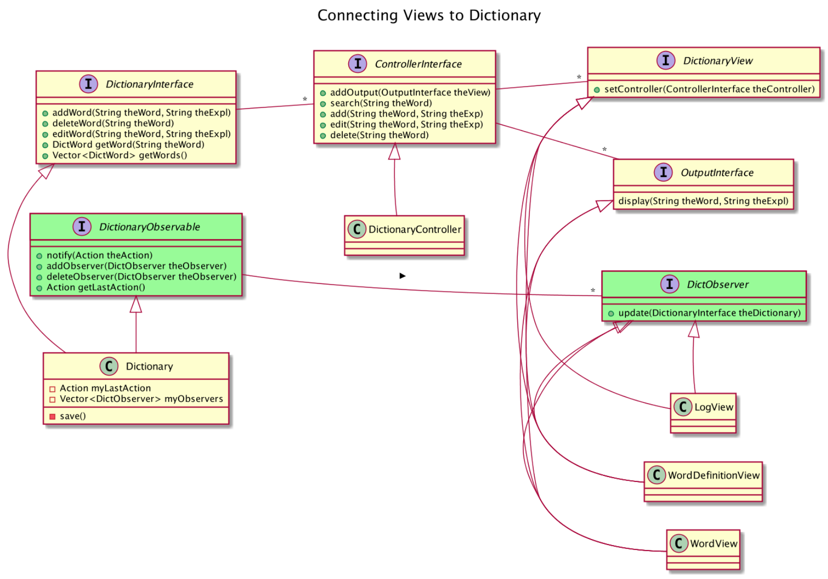

I
DictionaryInterface
addWord(String theWord, String theExpl)
deleteWord(String theWord)
editWord(String theWord, String theExpl)
DictWord getWord(String theWord)
Vector<DictWord> getWords()
*
I
ControllerInterface
addOutput(OutputInterface theView)
search(String theWord)
add(String theWord, String theExp)
edit(String theWord, String theExp)
delete(String theWord)
*
I
DictionaryView
setController(ControllerInterface theController)

2.4 Class Diagram II

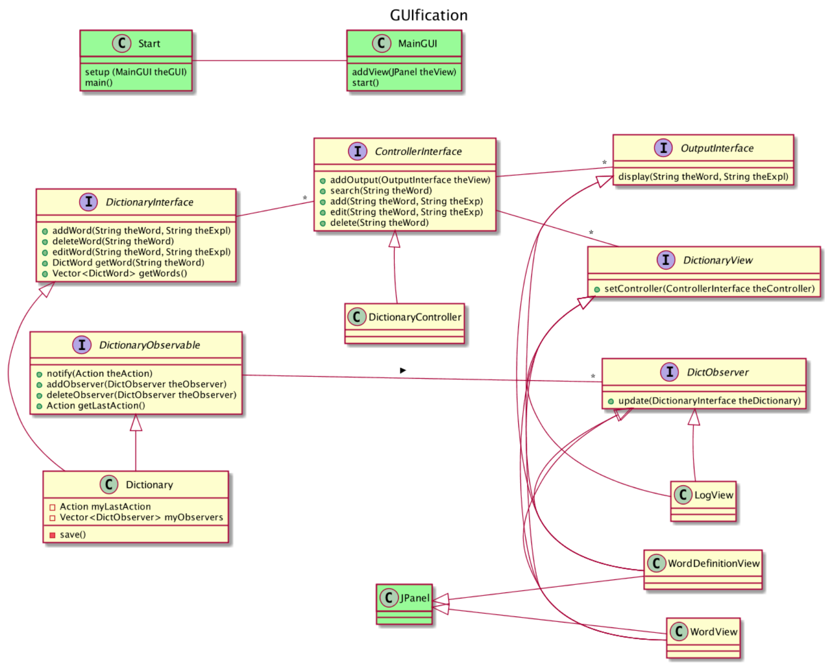
Concrete Implementations


I
DictionaryView
setController(ControllerInterface theController)
I
DictionaryInterface
addWord(String theWord, String theExpl)
deleteWord(String theWord)
editWord(String theWord, String theExpl)
DictWord getWord(String theWord)
Vector<DictWord> getWords()
*
I
ControllerInterface
addOutput(OutputInterface theView)
search(String theWord)
add(String theWord, String theExp)
edit(String theWord, String theExp)
delete(String theWord)
C
LogView
*
I
OutputInterface
display(String theWord, String theExpl)
C
Dictionary
save()
C
DictionaryController
C
WordDefinitionView
C
WordView

2.5 Class Diagram III


Connecting Views to Dictionary
DictionaryInterface
addWord(String theWord, String theExpl)
deleteWord(String theWord)
editWord(String theWord, String theExpl)
DictWord getWord(String theWord)
Vector<DictWord> getWords()
ControllerInterface
addOutput(OutputInterface theView)
search(String theWord)
add(String theWord, String theExp)
edit(String theWord, String theExp)
delete(String theWord)
DictionaryView
setController(ControllerInterface theController)
OutputInterface
display(String theWord, String theExpl)
DictionaryObservable
notify(Action theAction)
addObserver(DictObserver theObserver)
deleteObserver(DictObserver theObserver)
Action getLastAction()
DictionaryController
DictObserver
update(DictionaryInterface theDictionary)
Dictionary
Action myLastAction
Vector<DictObserver> myObservers
save()
LogView
WordDefinitionView
WordView
*

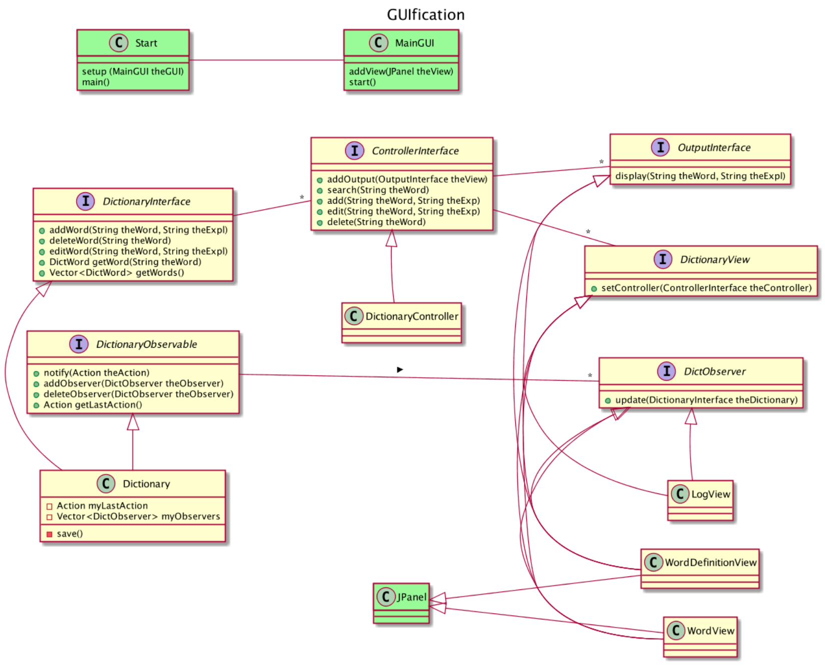
2.6 Class Diagram IV


GUIfication
Start
setup (MainGUI theGUI)
main()
MainGUI
addView(JPanel theView)
start()
DictionaryInterface
addWord(String theWord, String theExpl)
deleteWord(String theWord)
editWord(String theWord, String theExpl)
DictWord getWord(String theWord)
Vector<DictWord> getWords()
ControllerInterface
addOutput(OutputInterface theView)
search(String theWord)
add(String theWord, String theExp)
edit(String theWord, String theExp)
delete(String theWord)
OutputInterface
display(String theWord, String theExpl)
DictionaryView
setController(ControllerInterface theController)
DictionaryObservable
notify(Action theAction)
addObserver(DictObserver theObserver)
deleteObserver(DictObserver theObserver)
Action getLastAction()
DictionaryController
DictObserver
update(DictionaryInterface theDictionary)
Dictionary
Action myLastAction
Vector<DictObserver> myObservers
save()
LogView
WordDefinitionView
JPanel
WordView
*

2.7 Class Diagram: setup method

```
public static void setup(MainGUI theGUI) {
  // Create Dictionary
  Dictionary theDict = new Dictionary("dict.txt");
  debugDict(theDict); // Make sure there is stuff in it.

  // Create Views
  LogView lv=new LogView();
  WordView wv=new WordView();
  WordDefinitionView wdv=new WordDefinitionView();

  // Initialise views where necessary
  wv.getWords(theDict);

  // Create and Connect the Controller
  DictionaryController dc=new DictionaryController(theDict, wdv);
  lv.setController(dc);
  wv.setController(dc);
  wdv.setController(dc); // Circular, but ok

  // Add stuff to GUI
  // theGUI.addView(lv) // skip the LogView; it prints to console/file
  theGUI.addView(wv);
  theGUI.addView(wdv);

  // Connect views to dictionary, so that changes are reflected
  theDict.addObserver(lv);
  theDict.addObserver(wv);
  theDict.addObserver(wdv);
}
```

2.8 Discussion: Order of Implementation DISCUSSION


GUIfication
C Start
setup (MainGUI theGUI)
main()
C MainGUI
addView(JPanel theView)
start()
I ControllerInterface
addOutput(OutputInterface theView)
search(String theWord)
add(String theWord, String theExp)
edit(String theWord, String theExp)
delete(String theWord)
I OutputInterface
display(String theWord, String theExpl)
I DictionaryInterface
addWord(String theWord, String theExpl)
deleteWord(String theWord)
editWord(String theWord, String theExpl)
DictWord getWord(String theWord)
Vector<DictWord> getWords()
I DictionaryView
setController(ControllerInterface theController)
C DictionaryController
I DictionaryObservable
notify(Action theAction)
addObserver(DictObserver theObserver)
deleteObserver(DictObserver theObserver)
Action getLastAction()
I DictObserver
update(DictionaryInterface theDictionary)
C Dictionary
Action myLastAction
Vector<DictObserver> myObservers
save()
C LogView
C WordDefinitionView
C JPanel
C WordView
*
*
*
*