DOOM Case Study: Wedgi

- **Wedgi: Wordbench: Editing with a Graphical Interface**
  - towards a source browser? (see sniff+ [www.takefive.co.at](http://www.takefive.co.at))

- **Load files/URLs (other sources) into a framework that lets you edit the source, cut/copy/paste, etc. [what’s the etc.?]**
  - how might the user interact with the program?
  - What are the requirements? What are the use-cases?

- **Use-case**
  - textual, narrative description of the processes in a system
  - how the user interacts with the system (how components of the system interact with each other)
Wedgi: DOOM stage 1

- Get a working prototype up and running
  - work on requirements/specifications, prioritize project components, develop use-cases
  - choose core classes and responsibilities/relationships
  - write core stubs/headers, both behavior and state
  - get prototype up and running
- Load files/URLs (both initially?) navigate among sources
  - Readers for sources
  - CardLayout for navigation
  - Mediator, GUI, WedgiText, other classes?
  - How does cut/copy/paste work?
Multiple WedgiText objects

- How can we cycle between several documents loaded into Wedgi?
  - What operations might we perform on a document?
  - How are documents tracked internally?

- CardLayout
  - in theory facilitates “tabbed” layout, select among several displays
    - not all are actually shown (as should be in tabbed layout), in JDK 1.2 there is a tabbed layout
  - Objects added to container with CardLayout must be associated with a name
    - next, previous, first, last, and named object
    - What is name of first? What is name of next? Who tracks the names?
Alternatives to CardLayout for Wedgi

● Keep a vector/hashtable of WedgiTexts associated with a name
  ➤ how is view changed when user selects another WedgiText
  ➤ why doesn’t this work?
  ```java
  myWedge = selectedWedgi;
  validate();
  ```
  ➤ What can we do instead to get at the idea of swapping the displayed view for another? What are the issues?
    • What’s the state and who’s the parent

● How will save work, what about remove (for WedgiTexts)
  ➤ alternatives to maps of name -- WedgiText?
  ➤ What is a WedgiText? Compared to what?
DataTransfer

- Transferring objects and stuff between widgets in an application, between Java applications, between Java and other applications
  - application-specific clipboard, System clipboard
  - package objects and stuff up in some way that facilitates transfer
    - transfer in what format, out in another, e.g., HTML text
- Cut and Paste
  - put/get a `Transferable` (interface) to/from clipboard
    - `setContents(Transferable, Owner)`, `getContents(Owner)`
  - `Transferable` supports at least one `DataFlavor`
  - `DataFlavors` are built on MIME types
DataTransfer and DataFlavors

For more information see Core Java Vol. II and online AWT supplemental info

- **MIME: Multipurpose Internet Mail Extensions**
  - facilitate reading/writing attachments
  - can be used to facilitate MIME-aware Java app transfers

- **DataFlavors**
  - Java and other non-Java program: transfer text
  - Same java program: any object
  - MIME-aware inter-java transfer: text-encoded/MIME data

- **Transferable: returns string, stream or other source**

```java
DataFlavors[] flavors = trans.getTransferDataFlavors();
DataFlavor gifF = new DataFlavor("image/gif", "GIF bitmap");
if (trans.isDataFlavorSupported(gifF)) {
  Object o = trans.getTransferData(someFlavorObject);
}
```