Step 1: Develop database and usability requirements

The initial step in our design process is to conduct domain and task analyses. The gmaal of this step is to produce both database information and interactive system specifications. However, the

view [12]. The first of these, functional requirements, describes what the system should do. Our functional requirements include:

- Must provide a security mechanism to ensure that only authorized users submit data.
- Must guarantee reliability and completeness of the data, especially because much of it is user supplied.
- Must have a mechanism to distinguish published data from unpublished or pre-published data.
- Must allow submission of commonly published image types, including annotations.
- Must provide color reliability and reasonable resolution for images so that information is accurate enough to make science-based decisions.
- Must be accessible frog rnalalalbe lo18(e,27(s,(n)18(t)-24(s)7(i)n)18(t)-24gl)35 e s delroa19(e8(n)18(t)-2
  -

and to create a very simple interface for expressing queries in this subset. This required an extensive understanding of the domain and tasks.

Constraints Imposed by HTTP/HTML

The functional and performance requirements listed above derive from the characteristics and tasks of the zebrafish research community. Interface design for any Web-accessible application is

Step 2: Iterate detailed design process

Step 2 (Figure 1

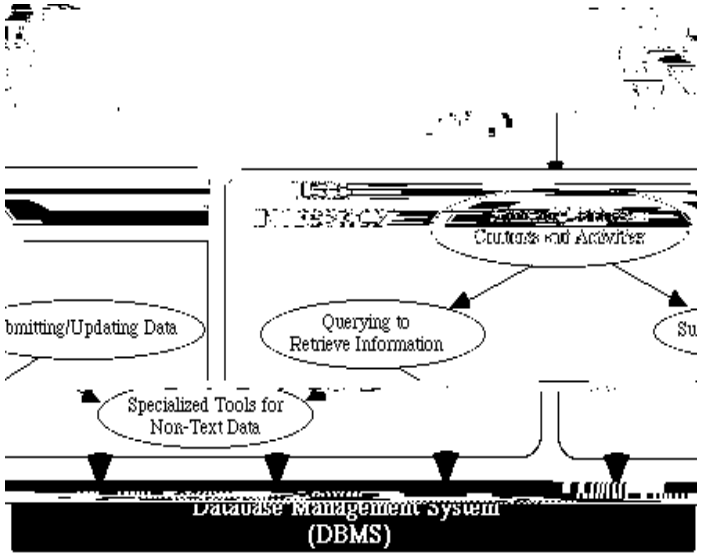

Figure 3: Four Interaction Modes supported by the ZFIN Interface. Arrows indicate the relationships of modes within the context of an interaction.

4.2.1 Surveying Database Contents and Activities

Our analysis of usability requirements dering Step 1 of the design process showed an

We implemented identification of individual users by using the "HTTP cookie" mechanism [11

of the data submission interface: supporting naive users, enforcing data security, supporting multi-step submissions, and providing context-sensitive navigation and help.

Supporting naive users.

We implemented this func tionality by saving inte tface state]TJı˝5.76 0 0 5.76 351.88 712.096 Tmı˝0 0 1 rgı˝0 T

4.2.4 Specialized Tools


element feedback
target data

Notes

1. This observation raises an obv(i)36(s)7ous question: if Java supports the full expressive power of a modern interface programming environment, why not implement the *entire* interface as a Java

Figure 4b: The ZFIN "home" page as seen by an auth.96 1gized submitter after l.96 1gging in.

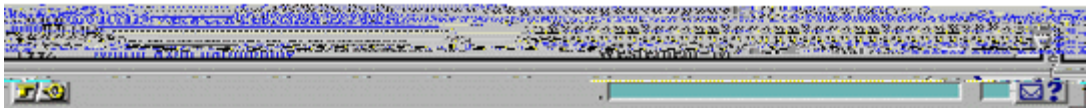

Figure 5b: Search interface for publications.

Figure 6a: A portion of the data entry form for submitting a new image to ZFIN.

Figure 7a: A typical digression sequence. To submit a new mutation, the user must specify the publication in which the mutation was first described.

Figure 7b: This brings up the constraint-based search interface, which is used to locate the publication record.

Figure 7c:

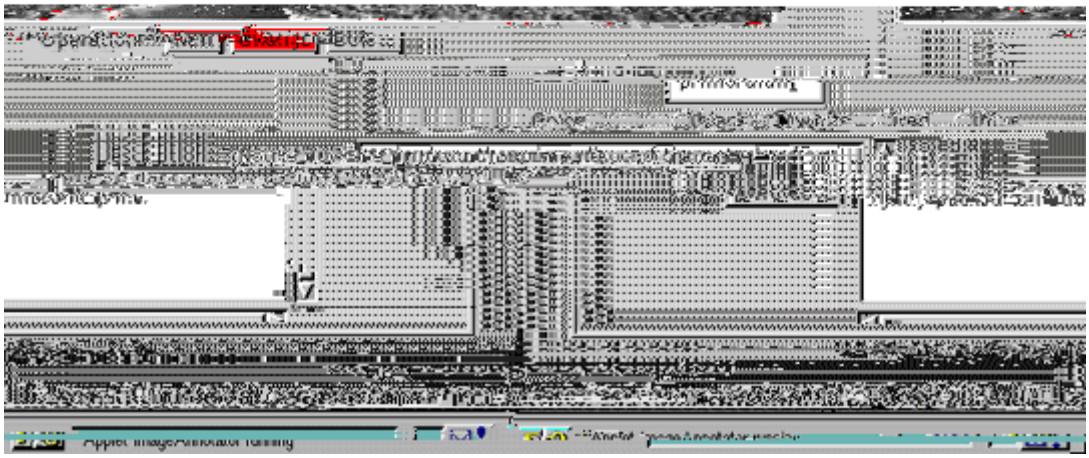

Figure 8a: The image annotator, shown in edit (versus viewing) mode. Annotations can be placed, edited and enhanced with arbitrary text descriptions.

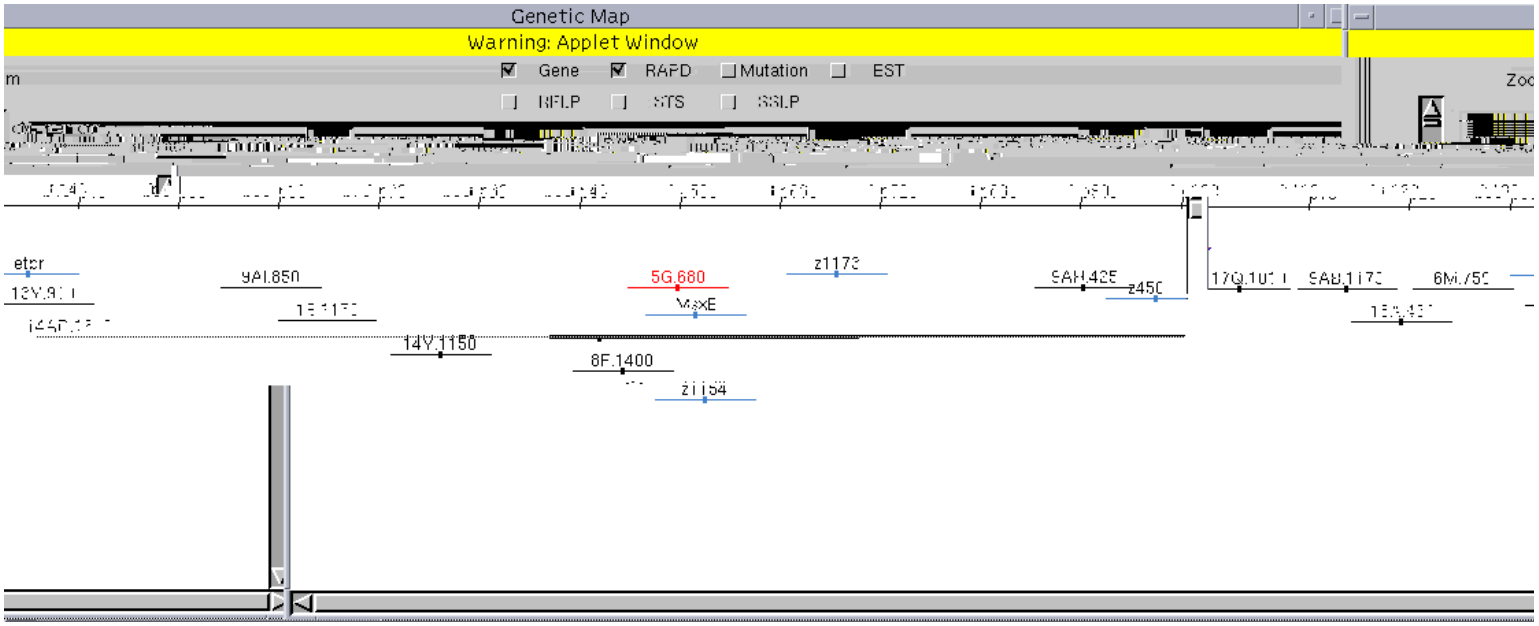
Genetic Map
Warning: Applet Window
Gene
RAPD
Mutation
EST
RFLP
STS
SSLP
etpr
9AI.850
5G.680
MaxE
z1173
z450
14Y.1150
8F.1400
z1154

Figure 8b: