A USER INTERFACE FOR THE GROUND OBSERVATIONS
DATABASE OF NASA'S DEEP IMPACT MISSION

Johnny Saucedo
Department of Information and Computer Science
University of Hawaii, Manoa
Honolulu, HI

ABSTRACT

NASA's Deep Impact mission will study the interior of a comet by excavating a crater more than 25 m deep and 100 m in diameter. The purpose of this project was to develop and implement a quality and efficient web-based user interface design to increase the functionality and practicality of the Deep Impact database to a broader group of scientists. The Deep Impact user interface was developed using the standard phases of a software life cycle: requirements, design, code, and testing. The first phase of development involved determining the requirements of the application. In the design phase of development, the application was modeled. The actual construction of the application began in the code phase. Finally, functional tests of the interface were presented to users in order to identify problems and unmet needs of the application. This usability testing was a critical step as it allowed the user to validate and evaluate the interface design and expose areas in which it required improvement. The results of the usability tests revealed that the improved interface was clear and efficient and provided a more comprehensive and useful tool than the previous interface.

INTRODUCTION

The Deep Impact Database at the Hawaii Institute for Astronomy contains observation data from ground telescopes used for calculations relating to NASA's Deep Impact mission, such as the rotation and orientation of the comet at impact.

A user interface is a display for the user that allows the user to interact with the system. An ideal user interface has two attributes: being visually apparent and making the user feel in control of the system. Previous to this project, a web interface to the Deep Impact database existed but it was extremely limited in its accessibility, was specific to a targeted group of users and was not considered to be a user-friendly tool for a broader group of users.

In addition to enhancing the previous search interface, a separate web application for editing data was also needed. Without this web application, in the past, the Hawaii Institute for Astronomy had to manually reinstall all data in the database even if only one entry needed to be edited. Reinstalling the data caused the database to be down for about four hours. Editing rights for the database is reserved only for the owner of the database, Karen Meech at the Hawaii Institute for Astronomy, thus the editing interface is a single user application.
All application development processes are broken down into distinct phases. These phases generally encompass the areas of determining requirements, developing design, and implementing and testing the application.

METHODS

The waterfall model for software engineering was used as the method for developing both the multi-user search application and single user editing application. In Figure 1, the double-border boxes represent the phases and the single-border boxes represent the product of a specific phase.

![Waterfall Model Diagram]

Figure 1. Software engineering's Waterfall model with feedback loop

Development Phase I: Requirements

The first phase of development involved determining the requirements of the application. To accomplish this, a thorough understanding of the structure and function of the database had to be acquired. The needs of the end user were assessed and all constraints and rules were defined. The specific benefits of the application, namely, increasing accessibility and decreasing server downtime, were also defined. After all requirements were gathered, it was determined that the scope of the project was to include a new set of search parameters and the ability to edit the data.

Development Phase II: Specification

The specification phase produced a description of the input and output. For both the search interface and the edit interface, the input was defined as the values entered by the user into the text boxes. User input was greatly limited by the use of client side validation, using regular expressions. The output for the search interface was a subset of the records in the database as well as a subset of the table attributes. The output for the
edit interface was a display of the record that was changed, as well as a message stating the change was successful.

Development Phase III: Design

In the design phase of development, the application was modeled. This was first accomplished on a conceptual level. The various data elements and functions were mapped out into flow charts and diagrams. The structure of the database was already in place and was not altered for this project; however, an analysis of the relationships between the tables was necessary, as the schema diagram did not accurately reflect the table structure relationships. The next level consisted of designing interface prototypes and the events that should occur as a result of the interaction between the user and the application. Next, the specific technology that was needed to develop the application was determined and the platform on which the design would be implemented was established.

Development Phase IV: Implementation

The actual construction of the application began in the implementation phase. Implementation produced an encoding of the design in a programming language to produce a working system. The output of the implementation phase was code written in Coldfusion and Structured Query Language.

Development Phase V: Testing

As the application was programmed, functional test interfaces were presented to users in order to identify problems and unmet needs of the application. This allowed for modification of the design as the application was being programmed, which is a more efficient process. Usability testing was a critical step as it allowed the user to validate and evaluate the interface design and expose areas in which improvement was required. It also served as a measure of how easy the interface is to use or how much instruction or guidance would be needed once the application were fully implemented.

Throughout all phases of development, the project will rely on the following resources: interviews with Deep Impact database designers and managers, interviews and continuous collaboration with interface users, specialized publications, and manuals relating to data contained in the Deep Impact database or the database itself.

One of the fundamental ideas in computer science and engineering is the use of layers of technology. The applications for this project were written in Coldfusion and Structured Query Language, to be run on a Unix server. Furthermore, the applications for the searching of data were written using the application framework called Fusebox. An application framework is a collection of software components that programmers use to build applications for the domain the framework addresses; it is also a set of best practices for organizing applications.
RESULTS

Figure 2 displays the additional functionality of the web interface. It is now possible to search for observations by institution, telescope, observatory, observer, operator, and/or instrument. Documentation in the form of a help page was added as well. A pop-up link on the top right hand side of the page explains the search parameters in detail, and gives the user information about the data, such as the range in values for search parameters.

Next to the ‘ObservatoryID’ search box in the interface is a hyperlink to a pop-up window which contains a list of observatory codes. This was provided to the user so that they do not have to look up the code in a book or on another website. There is also an additional “return file format” option of comma-separated values for data portability.

All new search parameters were added to the ‘Sort By’ option.

---

### Deep Impact - Data Query Form

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ObjID</td>
<td>(0,1,2,...,n)</td>
</tr>
<tr>
<td>JD</td>
<td>(Decimal)</td>
</tr>
<tr>
<td>UTMtime (mm/dd/yy)</td>
<td>From: [ ] To: [ ]</td>
</tr>
<tr>
<td>Filt (Decimal Format)</td>
<td>F B V R G I</td>
</tr>
<tr>
<td>Apert (A, arcsec)</td>
<td>Min: [ ] Max: [ ]</td>
</tr>
<tr>
<td>Vectors (Sun Centered)</td>
<td>Min: X Y Z VX VY VZ</td>
</tr>
<tr>
<td>Seeing</td>
<td></td>
</tr>
<tr>
<td>Logheader</td>
<td></td>
</tr>
<tr>
<td>String</td>
<td></td>
</tr>
<tr>
<td>Observatory ID</td>
<td>(example: 695) Observatory Codes</td>
</tr>
<tr>
<td>Observer ID</td>
<td>(example: SM HS)</td>
</tr>
<tr>
<td>Operator</td>
<td></td>
</tr>
<tr>
<td>Telescope</td>
<td></td>
</tr>
<tr>
<td>Sort by:</td>
<td>ObjID ASCENDING</td>
</tr>
</tbody>
</table>

![Figure 2. Deep Impact multi-user search interface](image-url)
Figure 3 displays the results of the waterfall model with the feedback method, including user feedback via user testing. The interface allows the user to edit magnitude, filter type, and error for any observation. The form contains validation logic, giving the user feedback if they did not enter a valid entry. This was done via a scripting language on the client-side so that the user does not have to wait for the server to respond, and the server does not have to do the extra work of validation.

Once the user enters the information for editing and presses the submit button, the users are shown the information they want to edit and are asked to confirm their decision to proceed with the update. If the update is completed, then the user is shown the edited data and is told that the edit was successful.

Figure 3. Deep Impact single-user edit interface
CONCLUSION

By using a web-based interface to disseminate information in an efficient and controlled way, the Deep Impact database will contribute to the field of astrobiology by enhancing the access, efficiency and productivity of scientists involved in the Deep Impact mission. Efficiently producing an interface that is visually apparent and gives the user a sense of control of the system was accomplished using the waterfall model with feedback in which usability testing was stressed. The addition of the editing web application increased the number of hours per day that the database was functional, thereby increasing accessibility to scientific data.

ACKNOWLEDGEMENTS

The author would like to thank NASA and the Hawaii Space Grant Consortium for providing undergraduates with the unique opportunity to work closely with researchers. The author would like to thank Dr. Kim Binsted for serving as his mentor throughout this research project. The author would also like to thank Karen Meech, Narayan Raja, and especially Jana Pittchava for being so patient and clear while explaining technical astronomy concepts and definitions.

REFERENCES


