ARMY RESEARCH LABORATORY

ARL

Graphical User Interface (GUI) Design for Ballistic Research Laboratory–Computer-Aided Design's (BRL–CAD's) Geometry Difference (GDiff) Tool

by Matthew Bayes

NOTICES

Disclaimers

The findings in this report are not to be construed as an official Department of the Army position unless so designated by other authorized documents.

Citation of manufacturer's or trade names does not constitute an official endorsement or approval of the use thereof.

Destroy this report when it is no longer needed. Do not return it to the originator.

Army Research Laboratory
Aberdeen Proving Ground, MD 21005

ARL-CR-0756   **April 2015**

Graphical User Interface (GUI) Design for Ballistic Research Laboratory–Computer-Aided Design's (BRL–CAD's) Geometry Difference (GDiff) Tool

Matthew Bayes
Survivability/Lethality Analysis Directorate, ARL

Prepared

Under contract

W911NF-10-2-0076

REPORT DOCUMENTATION PAGE			*Form Approved* *OMB No. 0704-0188*		

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing the burden, to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.
PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED (From - To)
April 2015	Contractor Report	06/2014–08/2014

4. TITLE AND SUBTITLE	5a. CONTRACT NUMBER
Graphical User Interface (GUI) Design for Ballistic Research Laboratory–Computer-Aided Design's (BRL–CAD's) Geometry Difference (GDiff) Tool	W911NF-10-2-0076
	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)	5d. PROJECT NUMBER
Matthew Bayes	
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)	8. PERFORMING ORGANIZATION REPORT NUMBER
US Army Research Laboratory ATTN: RDRL-SLB-S Aberdeen Proving Ground, MD 21005	ARL-CR-0756

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

Ballistic Research Laboratory–Computer-Aided Design (BRL–CAD) is a solid modeling system used by the Army for ballistics analysis. US Army Research Laboratory (ARL) target describers use the BRL–CAD Geometry Difference (GDiff) tool to determine the differences between their target models. GDiff is currently a command-line-only utility lacking a graphical user interface (GUI). To improve the usability of the tool and to increase target describers' productivity, a GUI for the tool is being created. Various differencing-type GUIs were researched to create an initial GUI design. Several possible designs were sketched and then made, and an initial design concept was reviewed by ARL target describers who provided feedback that improved the GUI design. A final sketch of the GUI was then created. The GUI is being implemented using the Tcl/Tk "tookit" in the form of an extension to Archer, which is a GUI for BRL–CAD itself. Previous plug-ins for Archer are being examined to determine how to best implement the new GUI that is being written and tested.

15. SUBJECT TERMS
BRL–CAD, DIFF, GUI, GDiff

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
A. Report	b. ABSTRACT	c. THIS PAGE	UU	16	Geoffrey Sauerborn
Unclassified	Unclassified	Unclassified			**19b. TELEPHONE NUMBER** *(Include area code)* 410-278-6178

Standard Form 298 (Rev. 8/98)
Prescribed by ANSI Std. Z39.18

Contents

List of Figures

Acknowledgments

Student Biography

Now entering his sophomore year of computer science at the University of Maryland, College Park, Matthew's goal is to earn a bachelor of science degree and then attend graduate school for research in artificial intelligence. Before this project at the US Army Research Laboratory (ARL), he worked on a senior capstone research project at the Science and Mathematics Academy at Aberdeen High School, Maryland. In that project, Matthew researched the computation of the number of unique states of a given size of ring-based cellular automata.

1. Introduction/Background

The Ballistic Research Laboratory–Computer Assisted Design (BRL–CAD) is an open-source, solid modeling system utilized by the US Army to create models for use in ballistics analysis. My project was to create a graphical user interface (GUI) for the command-line-based Geometry Difference (GDiff) tool to make it more efficient to use and much more user friendly. This task was important as it would allow the target describers[1] to more effectively determine the differences between models. This is necessary because more than one target describer might be working on the same model at the same time, and they would need to view the differences between each working version of the models. When I began my student summer research project at the US Army Research Laboratory (ARL), my computer access was not immediately available due to standard security procedures. So, I invested that time in getting the computer hardware set up and studying BRL–CAD-related documentation. I also was asked to independently set up video-digitization equipment intended to archive original recordings of significant importance relating to BRL–CAD and computing history. I proceeded to research and study the equipment provided determining how to best configure and make the system operational; this was accomplished in less than a day. Once I got everything functioning, I wrote instructions to be used by Army Science and Engineering Apprenticeship Program (SEAP) high-school students who were arriving the following week; this would enable them to immediately start digitizing the large set of tapes that showcase much of BRL–CAD's history.

Shortly after completing this task, I received ARL computer access. My next task was to compile BRL–CAD source code and complete multiple tutorials to become familiar with BRL–CAD's tools and utilities. Once I accomplished this, I was tasked with modeling a wooden mannequin that would later be used at an off-site activity I participated in. (See Fig. 1.) Modeling the mannequin helped me understand how BRL–CAD worked. Once I was finished with this, I began to acquaint myself with the BRL–CAD source code. Learning how to navigate such a large code base (more than 1.5 million lines of code) was a very interesting experience.
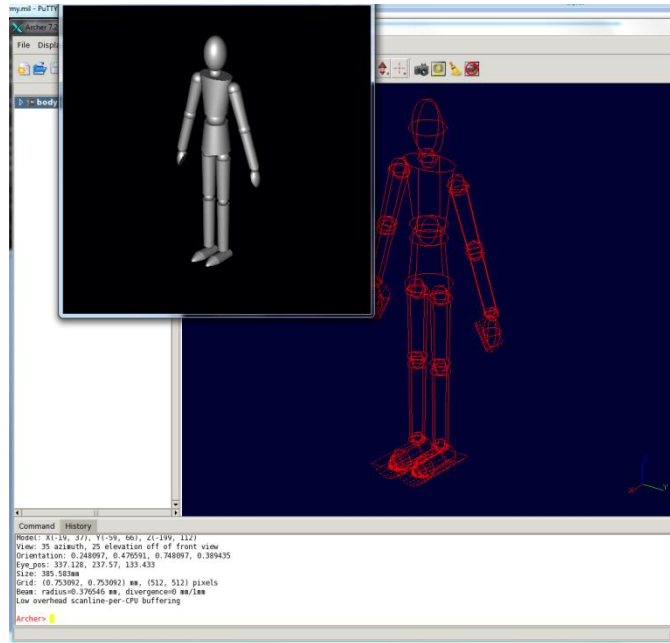
Fig. 1   BRL–CAD mannequin

2.   Experiment/Calculations

Once assigned to my main project, I began researching the designs of various geometry-differencing tools similar to GDiff in order to grasp what is expected from one. Once I had a sufficient amount of background knowledge on GDiff type tools, I began creating different drafts of possible GUIs for the BRL–CAD GDiff tool. After designing a number of concept sketches, I compiled the best features from each into a single design. This initial design consisted of one main window that contained a single-object view, a hierarchy listing for the currently selected model, and buttons to change between models and various views. This design would allow the user to load 2 or 3 models to compare. The object view would allow the user to button-select which model they would like to view, as well as the way they would like to view the differences between the models. The hierarchy would be displayed for any model selected. This was the design to be presented to ARL's target describers. Target describers are modeling experts and would be the tool's primary users.

When I met with the target describers, they thought the design that I came up with was very good. They proved valuable feedback leading me to add a few new features such as a summary window to the GUI design. I used all of the feedback toward a final design for the GUI. One of the most significant changes requested was to focus on diffing 2 models, as opposed to simultaneously doffing 3, which is rarely done. Another change included adding an initial open

dialog that would come up as soon as the tool was launched. This allows the user to select which geometries that they would like to compare. A summary window was also added that would launch as soon as the user selected the files to be loaded. This window (Fig. 2) gave a brief overview about the differences and then gave the traditional text-based listing of the differences. From this summary window, the user was given the option to either launch the main window of the GUI or to quit the tool if they believed they had a sufficient amount of information about the differences of the 2 models. Once I had this final design created, I began implementing the GUI.

Display 1st object, 2nd object, or differences respectively

File

File Menu
Open
Help
Summary
Exit

1 2 D

Hierarchy

Object View

A B R

Display geometry added to first to get second, display both added and removed geometry, and display geometry removed from first to get to second respectively.
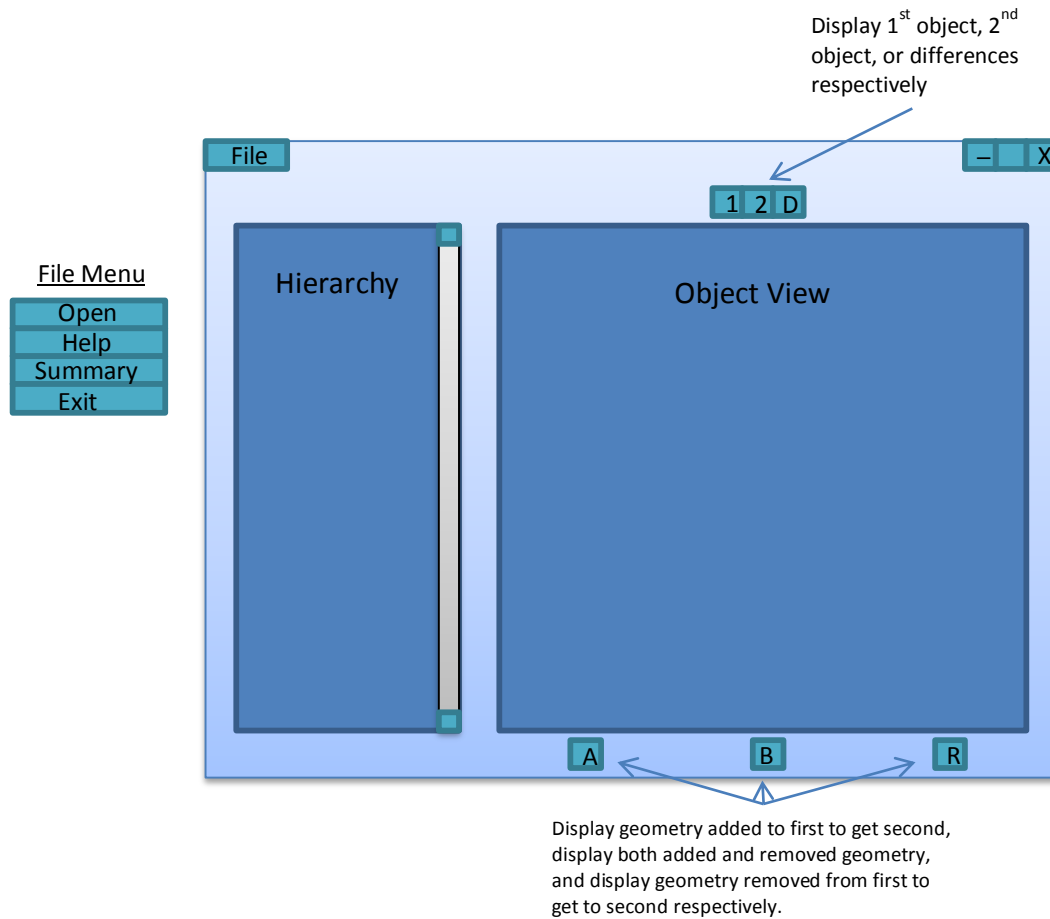
Fig. 2   Main GUI window mock-up

The GUI was being created in the *incr Tcl/Tk* language that the rest of Archer was implemented in. This posed multiple challenges for me. I was not familiar with Tcl/Tk or the object-oriented version of it (*incr Tcl/Tk*). I started studying these languages so I could successfully complete my project. This was harder for me than expected. Once I had a sufficient foundation in the language, I began implementing my GUI. First, I had to figure out how to launch the GUI as an extension to Archer. I thought I had a solid idea on how to do this, so I started writing the GUI in *incr Tcl/Tk*. After writing some initial parts of the GUI, I wanted to test it. This is when I attempted to make it launch from Archer. After several attempts, I could not get it to work.

Because of this, I decided to begin implementing the GUI in Tcl/Tk; this way, I could launch it directly from the command line, which would allow testing the GUI during creation. This allowed me to see the initial GUI "load window" launch followed by the summary window as intended. While beginning to implement the main window, though, I encountered a major problem. At this point it seems I will need *incr Tcl/Tk* code to implement any type of rendering window or hierarchy. This is a setback, as I will have to get the code working in *incr Tcl/Tk*, which was the original problem. I am currently researching how to accomplish this.

Meanwhile, I also participated in an off-site work session that supported the Joint Trauma Analysis and Prevention of Injury in Combat program. The goal of this off-site session was to overhaul the joint-constraint system to be able to manipulate joints in real time using a computer's mouse. My main task was to create an "object edit" panel in Archer for the joint primitive. I successfully implemented this, which allowed the editing of the joint constraints to be more user friendly.

3. Results and Discussion

The GUI, in its current state, has a functioning load window and summary window. Its main window is not currently functioning due to my as-yet incomplete understanding of *incr Tcl/Tk* and inability—so far—to locate features of Archer that would be used in the GUI's implementation. I am currently working toward a functional main window that would provide at least a basic level of utilization.

4. Summary and Conclusions

This summer at ARL was a good learning experience for me—even if I did not accomplish everything that I wanted to. Before I came here, I had no knowledge of the BRL–CAD source code or even what BRL–CAD actually was. I had never even heard of Tcl/Tk or worked much in the coding of GUIs. I had never worked on a project with such a large code base. I did not have much experience with the Linux operating system or developing on it. I managed to learn about all of these things this summer. I believe that this newfound knowledge will be incredibly beneficial to me throughout the rest of my career, both academic and in the workforce. I also spent much of my time working alongside the SEAP students and was able to assist them in resolving computer-science-related problems. It was good working with others, as well as just figuring out some of their problems; some problems were quite challenging. Even though this did take quite a part of my time, I believe that it was worth it.

5. Notes

1. Target describers are CAD experts who are able to generate detailed descriptions of military systems suitable for threat analysis. These "targets" could be generated from "scratch" using blueprints or other original source data describing the system. More often, military-system manufacturers' own CAD designs are imported and modified until they are suitable for vulnerability/lethality analysis.

List of Symbols, Abbreviations, and Acronyms

ARL   US Army Research Laboratory

BRL–CAD   Ballistic Research Laboratory–Computer-Aided Design

GDiff   Geometry Difference

GUI   graphical user interface

SEAP   Science and Engineering Apprenticeship Program (Army)

1 (PDF)	DEFENSE TECHNICAL INFORMATION CTR DTIC OCA
2 (PDF)	DIRECTOR US ARMY RSRCH LAB RDRL CIO LL IMAL HRA MAIL & RECORDS MGMT
1 (PDF)	USARL RDRL SLE R FLORES

ABERDEEN PROVING GROUND

1 (HC)	DIR US ARMY EVALUATION CTR HQ TEAE SV P A THOMPSON 2202 ABERDEEN BLVD 2ND FL APG MD 21005-5001
4 (2 HC 2 PDF)	DIR USARL RDRL SL J BEILFUSS (1 HC, 1 PDF) P TANENBAUM (1 HC, 1 PDF)
6 (PDF)	DIRECTOR US ARMY RSRCH LAB RDRL SLB S M PERRY W BOWMAN S MORRISON N REED G SAUERBORN C YAPP
1 (PDF)	DIRECTOR US ARMY RSRCH LAB RDRL SLB G S HORNUNG
1 (PDF)	DIRECTOR US ARMY RSRCH LAB RDRL SLB W C KENNEDY

INTENTIONALLY LEFT BLANK.