The GEV Display Inspector/Editor

Gordon S. Novak Jr.
Heuristic Programming Project
Computer Science Department
Stanford University
Stanford, CA 94303

Abstract

The GLISP Display Inspector/Editor, GEV, is an interactive program which runs on Lisp machines with high-resolution displays. It uses GLISP structure descriptions to interpret and intelligently display Lisp data. Using the display "mouse", the user can move from one object to another, and can "zoom in" on objects of interest. The editor keeps track of the types of the data being examined; the user can examine the specified type of the data, and can edit the data using the Lisp editor or a type-specific editor. Computed properties of objects can be displayed automatically or on demand. While based on GLISP, the system can also be used on any Lisp data without requiring that the GLISP compiler be loaded.

1. Introduction

The GLISP Display Inspector/Editor, GEV\(^1\), is a knowledge-based interactive program which allows the user to display, inspect, and edit structured data on a Lisp machine with high-resolution display. GEV gets its power from descriptions of the data to be examined; GEV remembers the types of all data as a structure is traversed, and uses this knowledge to generate intelligent displays.

Data structures are described to GEV using the GLISP Structure Description language \([1]\); if the user is already programming in GLISP, nothing extra is needed to use GEV. However, it is easy to describe Lisp structures to GEV; with some restrictions, GEV can be used independently of GLISP, without requiring that

\(^1\)GEV, for "GLISP Edit Value", is the name of the function which invokes the editor.
the GLISP compiler be loaded.

2. Invoking GEV

The GEV editor is invoked by giving it a value and a GLISP description of the value\(^2\), e.g.:

\[
(GEV \ C \ 'CIRCLE)
\]

This function call requests editing of the value of the variable C, which is to be interpreted as an instance of a CIRCLE. GEV will respond by creating a window and displaying the components of the circle C within the window.

\(^2\)For GLISP Objects, the description is automatically retrieved, and may be omitted.
The GEV window is divided into three sections:

1. The edit path. This section provides an abbreviated display of the path via which the currently examined object was reached from the object upon which GEV was initially invoked.

2. The actual data contained in the object being examined.

3. Computed properties of the object being examined.

Within each section, data are displayed in two columns. The name of each data item (either its variable name or substructure name) is displayed on the left, and the value of the data item is displayed on the right. Each data item occupies one line on the display. Often, the value of the datum cannot be fully displayed; in such cases, a shortvalue of the datum is displayed, and a tilde (~) is displayed to the left of the value to indicate that it is an approximation of the real value.

A menu of commands is displayed below the edit window.

3. Interaction with GEV

Most interaction with the GEV program is done using the display mouse; commands are selected using the Left mouse button. The mouse can be used to select items in either the edit window or the command menu. If a name field in the edit window is selected, the name and its declared type are printed in the typescript window. If a value field in the edit window is selected, the editor will perform a "push" to that value (or a "pop" to it if the selected value is in the Edit Path section), either taking it as a new object to be displayed, or displaying it in greater detail. An attempt to select the value of an item which is declared to be of a simple type (ATOM, INTEGER, etc.) will be ignored. When a new value item is selected, an additional line is added to the Edit Path section, and the selected value becomes the currently edited object.

4. Command Selection

A command menu below the edit window allows the user to select commands to GEV. These commands are:

1. QUIT: Terminate GEV and close its window. When a name or value item has been selected (indicated by inversion of the item on the display), the QUIT command may be used to cancel the selection without terminating GEV.

2. POP: Pop up from the currently examined item to its predecessor in the edit chain. A POP from

---

3At the present time, GEV is implemented only for the Xerox D-machine family.

4Note, however, that a datum whose stored form is a simple type can be "overloaded" as a structured type. For example, a datum whose stored type is REAL could be declared to be of type RADIANS, with associated computed properties such as DEGREES.
the top item causes a QUIT.

3. EDIT: Invoke the Editor on the currently examined item. The editor may be a special one specified for the datatype, or the system editor.

4. PROGRAM: Interactively acquire a program to be performed on the current object, and execute it.

5. PROP, ADJ, ISA, MSG: Display a menu of PROPERTIES, ADJECTIVES, ISA-adjectives, and Messages for the type of the current item.

4.1. Property Selection

When PROP, ADJ, ISA, or MSG is selected, GEV will display a menu of the names of the properties of the appropriate type which are declared in the GLISP Structure Description for the type of the current object. In addition, a "Quit" option is provided in the menu. For PROPERTIES, an "All" option is provided to display all properties if there are more than one; in addition, the property "self" is always defined, and will cause the actual Lisp value of the current item to be prettyprinted in the typescript window.

4.2. Program Specification

When the PROGRAM command is selected, GEV will acquire interactively (through menu selections) a program to be performed on the current object. The programs which are available are looping programs which operate on a list of items which is visible from the current object. The first menu which is presented to the user when the PROGRAM command is given is a list of Operations; the operations provided are COLLECT, TOTAL, AVERAGE, MAXIMUM, and MINIMUM. Once an operation has been selected, the Set over which the operation operates is selected; the menu will show all data items of the current object which are represented as lists. Finally, menus are presented to allow the user to select the datum from within the list element of the selected list. For the COLLECT operation, any data may be collected; for the other operations, only numeric data or structured data which may contain numeric data are presented. Menus are presented to "walk through" the selected data until either a basic data type is reached or (for COLLECT) the "Done" menu item is selected. GEV then creates a GLISP program to perform the specified operations, calls GLISP to compile it, and executes it on the current object. The result is printed in the typescript window and added to the GEV edit window. An example of a program specification and its result are illustrated below.
5. Object Declarations for GEV

A few PROPERTIES may be defined for an object type to control the displays produced by GEV. The property SHORTVALUE, if defined, will be used to display an object from a higher level; for example, the SHORTVALUE for a PERSON record could be defined as the last name of the person. Defining SHORTVALUES is important to make higher-level displays meaningful.
The property DISPLAYPROPS, if defined, causes certain computed PROPERTIES to be displayed automatically when an object of that type is visited. The value of DISPLAYPROPS should be a list of atomic PROPERTY names, or T to display all properties.

The message EDIT, if defined for an object type, will be invoked in response to an EDIT command; otherwise, the Lisp editor is used.

6. GEV and GLISP
GEV will automatically generate and compile functions to compute properties which are specified as GLISP expressions; this makes it very easy to build an "office automation" system based on a set of object descriptions and GLISP functions and expressions.

However, GEV can be used without the GLISP compiler being present if desired. In order to do so, the following restrictions must be observed:

1. All PROPERTY, ADJ, ISA, and MSG definitions given for objects must specify Lisp function names rather than GLISP expressions, since the GLISP compiler is required for expressions. In addition, a RESULT type should be specified for each function:

   PROP ((TOTAL TOTALFN RESULT REAL)  
   (SHORTVALUE TOTALFN RESULT REAL)  
   (DISPLAYPROPS TRUEFN)  
   ...)

2. The auxiliary file GEVSTR must be loaded to interpret structure descriptions.

3. The PROGRAM feature is not available.

7. Example File
A file containing GLISP object descriptions and a small amount of data\(^5\) is included as an appendix of this report. This file is useful as a set of examples of the declarations needed for GEV. The file may also be loaded as a test file for testing the use of GEV.

---

\(^5\) Much of the data is artificial.
References

[1] Novak, G. S.
changes to: (FNS GEVDEMO-INIT)
(VARS GEVDEMOCOMS)

previous date: "26-OCT-82 16:10:02" (DSK)GEVDEMO.LSP;20)

(PRETTYCOMPRINT GEVDEMOCOMS)
(RPAQQ GEVDEMOCOMS ((GLISPOBJECTS PROJECT CONTRACT AGENCY PERSON BUDGET ADDRESS PHONE-NUMBER DATE PICTURE CAMPUS-ADDRESS BUILDING CIRCLE VECTOR RADIANS DEGREES RVECTOR)
(FNS GEVDEMO-INIT TODAYS-DATE TOTAL-BUDGET)
(PROP GLRESULTTYPE TODAYS-DATE)
(P (GEVDEMO-INIT)))

[GLISPOBJECTS

(PROJECT

[ATOM (PROplist (TITLE STRING))
 (ABBREVIATION ATOM)
 (ADMINISTRATOR PERSON)
 (CONTRACTS (LISTOF CONTRACT))
 (EXECUTIVES (LISTOF PERSON))

PROP ((SHORTVALUE (ABBREVIATION)))
 (DISPLAYPROPS (T))
 (BUDGET TOTAL-BUDGET))

(CONTRACT

[ATOM (PROplist (TITLE STRING))
 (LEADER PERSON)
 (SPONSOR AGENCY)
 (BUDGET BUDGET))

PROP ((SHORTVALUE (TITLE)))

(AGENCY

[ATOM (PROplist (NAME STRING))
 (ABBREVIATION ATOM)
 (ADDRESS ADDRESS)
 (PHONE PHONE-NUMBER))

PROP ((SHORTVALUE (ABBREVIATION)))

(PERSON

[ATOM (PROplist (NAME STRING))
 (INITIALS ATOM)
 (TITLE ATOM)
 (PROJECT PROJECT)
 (SALARY REAL)
 (SSNO INTEGER)
 (BIRTHDATE DATE)
 (PHONE PHONE-NUMBER)
 (OFFICE CAMPUS-ADDRESS)
 (HOME-ADDRESS ADDRESS)
 (HOME-PHONE PHONE-NUMBER)
 (PICTURE PICTURE))

PROP ((SHORTVALUE (INITIALS)))
 (CONTRACTS ((THOSE CONTRACTS OF PROJECT WITH LEADER=SELF)))
 (AGE ((THE YEAR OF (TODAYS-DATE))
 - BIRTHDATE:YEAR))
 (MONTHLY-SALARY (SALARY/12))
 (DISPLAYPROPS (T)))

ADJ [(FACULTY ((MEMB TITLE (QUOTE (PROF ASSOC-PROF ASST-PROF) )

```
```
(BUDGET)
  (LIST (LABOR REAL)
         (COMPUTER REAL))
  PROP ((OVERHEAD (LABOR*0.59))
         (TOTAL (LABOR+OVERHEAD+COMPUTER))
         (SHORTVALUE (TOTAL))
         (DISPLAYPROPS (T)))

(ADDRESS)
  (LIST (STREET STRING)
         (CITY STRING)
         (STATE ATOM)
         (ZIP INTEGER))
  PROP [(SHORTVALUE ((CONCAT CITY ", " STATE) )]

(PHONE-NUMBER)
  (LIST (AREA INTEGER)
         (NUMBER INTEGER))
  PROP [(SHORTVALUE ((CONCAT "/ AREA ") (SUBSTRING NUMBER 1 3)
                     " "
                     (SUBSTRING NUMBER 4 7)]
  ADJ ((LOCAL (AREA=415 OR AREA=408)))

(DATE)
  (LIST (MONTH INTEGER)
         (DAY INTEGER)
         (SHORTYEAR INTEGER))
  PROP [[MONTHNAME ((CAR (NTH (QUOTE (January February March April May June July August September
                         October November December))
                      MONTH]
                     (YEAR (SHORTYEAR + 1900)
                     (SHORTVALUE ((CONCAT MONTHNAME " " DAY ", " YEAR) )]

(PICTURE)
  ANYTHING
  MSG ((EDIT PAINTW)
        (GEVDISPLAY PICTURE-GEVDISPLAY))

(CAMPUS-ADDRESS)
  (LIST (BUILDING BUILDING)
         (ROOM ATOM))
  PROP [(SHORTVALUE ((CONCAT BUILDING:ABBREVIATION ", " ROOM) )]

(BUILDING)
  (ATOM (PROPLIST (ABBREVIATION ATOM)
                (NAME STRING)
                (NUMBER INTEGER))
    PROP ((SHORTVALUE (NAME))))

(CIRCLE)
  (LIST (START VECTOR)
         (RADIUS REAL))
  PROP [(PI (3.141593))
        (DIAMETER (RADIUS*2))
        (CIRCUMFERENCE (PI*DIAMETER))
        (AREA (PI*RADIUS^2))
        (SQUARESIDE ((SORT AREA)))
        (DISPLAYPROPS ((QUOTE (DIAMETER CIRCUMFERENCE AREA)])
MSG ((GROW (AREA++100))
(Shrink (AREA=AREA/2)))
(STANDARD (AREA=100.0)))

ADJ ((BIG (AREA>100))
(SMALL (AREA<80))) )

(VECTOR

(List (X INTEGER)
(Y INTEGER))

PROP [((MAGNITUDE ((Sqrt X+2 + Y+2)))
(ANGLE ((ArcTAN2 Y X T))
RESULT RADIANS)
(UNITVECTOR ((A RVECTOR WITH X = X/MAGNITUDE , Y = Y/MAGNITUDE)
ADJ ((ZERO (X IS ZERO AND Y IS ZERO))
(NORMALIZED (MAGNITUDE = 1.0)))

MSG [[(PRIN1 "(")
(PRIN1 X)
(PRIN1 ",")
(PRIN1 Y)
(PRIN1 ")")]]
(PRINT (+ self PRIN1)
(TERPRI)

(RADIANS

REAL

PROP [((DEGREES (self* (180.0/3.1415926))
RESULT DEGREES)
(DISPLAYPROPS (T))) ]

(DEGREES

REAL

PROP [((RADIANS (self* (3.1415926/180.0))
RESULT RADIANS)
(DISPLAYPROPS (T))) ]

(RVECTOR

(List (X REAL)
(Y REAL))

(SUPERS (VECTOR) )
)

(DEFINEQ

(GEVDEMO-INIT
[GLAMBDA NIL

(* edited: "6-NOV-82 14:41")

(* Initialize data structures for GEV demo.)

(PROG NIL

(HPP +(A PROJECT WITH TITLE = "Heuristic Programming Project" , ABBREVIATION = (QUOTE HPP)))

(MJH +(A BUILDING WITH ABBREVIATION = (QUOTE MJH)
  NAME = "Margaret Jacks Hall" , NUMBER = 460))

(ARPA +(AN AGENCY WITH NAME = "Defense Advanced Research Projects Agency" ,
  ABBREVIATION = (QUOTE ARPA)
  ADDRESS +(AN ADDRESS WITH STREET = "1400 Wilson Blvd." , CITY = "Arlington"
  STATE = (QUOTE VA)
  ZIP = 22209)
  PHONE +(A PHONE-NUMBER WITH AREA = 202 , NUMBER = 6944349))

(NSF +(AN AGENCY WITH NAME = "National Science Foundation" , ABBREVIATION = (QUOTE NSF)
  ADDRESS +(AN ADDRESS WITH STREET = "1800 G STREET N.W." , CITY = "Washington"
  STATE = (QUOTE DC)
(TODAYS-DATE
  (GLAMBDA NIL
    (A DATE WITH MONTH = 10, DAY = 15, SHORTYEAR = 82)))

(TOTAL-BUDGET
  (GLAMBDA (P:PROJECT)
    (PROG (SUM)
      (SUM+0.0)
      (FOR EACH CONTRACT SUM+<BUDGET:TOTAL>
        (RETURN SUM)))
    )
  )

(PUTPROPS TODAYS-DATE GLRESULTTYPE DATE)
(GEVDEMO-INIT)
(DECLARE: DONTCOPY
  (FILEMAP (NIL (4061 9998 (GEVDEMO-INIT 4071 . 9592) (TODAYS-DATE 9594 . 9764) (TOTAL-BUDGET 9766 . 9996))))
  STOP)