NASA-CR-192821

Exploratory Research for the Development of a Computer Aided Software Design Environment with the Software Technology Program

Progress Report
and
Summary Report

5/89 - 4/91
Charles Hardwick
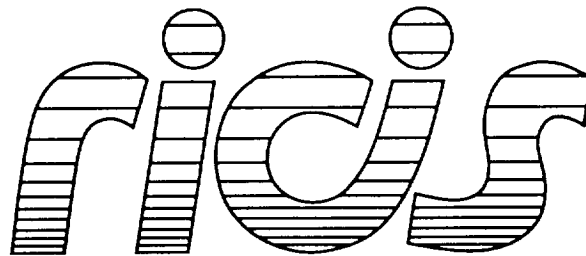
Cooperative Agreement NCC 9-16
Research Activity No. SE.23

NASA Johnson Space Center
Mission Support Directorate

Research Institute for Computing and Information Systems
University of Houston-Clear Lake

The RICIS Concept

The University of Houston-Clear Lake established the Research Institute for Computing and Information Systems (RICIS) in 1986 to encourage the NASA Johnson Space Center (JSC) and local industry to actively support research in the computing and information sciences. As part of this endeavor, UHCL proposed a partnership with JSC to jointly define and manage an integrated program of research in advanced data processing technology needed for JSC's main missions, including administrative, engineering and science responsibilities. JSC agreed and entered into a continuing cooperative agreement with UHCL beginning in May 1986, to jointly plan and execute such research through RICIS. Additionally, under Cooperative Agreement NCC 9-16, computing and educational facilities are shared by the two institutions to conduct the research.

The UHCL/RICIS mission is to conduct, coordinate, and disseminate research and professional level education in computing and information systems to serve the needs of the government, industry, community and academia. RICIS combines resources of UHCL and its gateway affiliates to research and develop materials, prototypes and publications on topics of mutual interest to its sponsors and researchers. Within UHCL, the mission is being implemented through interdisciplinary involvement of faculty and students from each of the four schools: Business and Public Administration, Education, Human Sciences and Humanities, and Natural and Applied Sciences. RICIS also collaborates with industry in a companion program. This program is focused on serving the research and advanced development needs of industry.

Moreover, UHCL established relationships with other universities and research organizations, having common research interests, to provide additional sources of expertise to conduct needed research. For example, UHCL has entered into a special partnership with Texas A&M University to help oversee RICIS research and education programs, while other research organizations are involved via the "gateway" concept.

A major role of RICIS then is to find the best match of sponsors, researchers and research objectives to advance knowledge in the computing and information sciences. RICIS, working jointly with its sponsors, advises on research needs, recommends principals for conducting the research, provides technical and administrative support to coordinate the research and integrates technical results into the goals of UHCL, NASA/JSC and industry.

RICIS Preface

The views and conclusions contained in this report are those of the author and should not be interpreted as representative of the official policies, either express or implied, of UHCL, RICIS, NASA or the United States Government.

Research Activity
SE.23

Progress Report

Period:

        May 1989 - February 1991

Status:

        Summary of the work done:

        Phase 1 of the contract was initiated.

        Field studies were conducted by MCC to determine areas of research of mutual interest
to MCC and JSC.

        NASA personnel from the Information Systems Directorate and research faculty from
UHCL/RICIS visited MCC in Austin, Texas to examine tools and applications under
development in the MCC Software Technology Program.

        MCC personnel presented workshops in hypermedia, design knowledge capture and
design recovery on site at JSC for ISD personnel.

        The following programs were installed on workstations in the Software Technology Lab,
Building 12, NASA/JSC:
            GERM (Graphic Entity Relations Modeler)
            gIBIS (Graphic Issues Based Information System)
            DESIRE (Design Recovery tool)

        These applications were made available to NASA for inspection and evaluation.

PROBLEMS:

        Programs developed in the MCC Software Technology Program run on the SUN
workstation. The programs do not require special configuration, but they will require
larger than usual amounts of disk space and RAM to operate properly.

PROJECTED ACTIVITIES:

        Each program will be available for research in the Software Technology Lab, Building
12. Personnel at NASA/JSC designing and developing systems software will have an
opportunity to work with the MCC programs in an effort to match solutions to problems
that exist or are anticipated at NASA.

Exploratory Research for the Development of a Computer Aided Software Design Environment with the Software Technology Program (STP)

SE.23

Summary Report, April 1991

Under the terms of the Phase 1 contract, Microelectronics and Computer Technology Corporation (MCC) delivered to NASA/JSC binary versions of software under development at MCC. The software was installed in computers in the Software Technology Lab of the Information Systems Directorate, Building 12.

The purpose of the Phase 1 activity was to allow ISD personnel to work with MCC software development tools in an effort to determine their usefulness in solving problems within the NASA/JSC software design environment.

Following a site visits by Bryan Fugate of MCC, the areas of design knowledge capture, graphical user interface, and design recovery were identified as being of mutual interest to JSC and MCC. Alan Plumb, Charles Pitman, Robert MacDonald, and Ken Crouse from NASA, and Chris Dede, Sharon Perkins and Charles Hardwick, from UHCL/RICIS, visited MCC in Austin, Texas to examine software design tools under development in the Software Technology Program (STP). In turn, Jeff Conklin, Dallas Webster, and Noreen Garrison presented workshops and demonstrations of GERM. gIBIS, and DESIRE on site at JSC.

DESIRE, gIBIS, and GERM, MCC tools addressing these requirements, were installed in Building 12 at JSC. The programs are being evaluated by personnel in the Information Systems Directorate, Software Technology Program.

gIBIS, a graphical issues based information system, and GERM, a hypermedia authoring program, were also installed at UHCL in the Research Computing and Data Facility (RCDF). Both programs are being evaluated for use in Design Knowledge Capture, a requirement for Space Station Freedom.

During the three month period from February 1991 and April 1991, MCC, JSC and RICIS personnel were cooperatively involved in assessing and evaluating the MCC applications installed in the JSC Software Technology Lab, and in the RCDF at the University of Houston-Clear Lake. MCC provided training sessions on the installed applications in Austin, Texas and at JSC.

Jeff Conklin and Dallas Webster of MCC made presentations and provided training support on gIBIS, GERM and DESIRE. Noreen Garrison of MCC made two visits to JSC to provide training on the use of GERM. In turn, Charles Hardwick (UHCL/RICIS), Ken Crouse (JSC/Engineering), and Alan Plumb (JSC/ISD) visited MCC in Austin, attended workshops, and worked with MCC personnel to gain a better understanding of the use and capabilities of the applications under review.

As a result of the evaluations, the Software Technology Branch expressed no interested in acquiring GERM, or in working with MCC to incorporate it into the NASA problems solving environment. (See Attachments.)

ISD expressed an interest in a further evaluation of DESIRE, and in working with MCC to develop a design recovery tool useful in recovering FORTRAN code. A follow-on task was defined to meet this research objective..

(Attachments)

AN ANALYSIS OF THE POTENTIAL USEFULNESS TO THE SOFTWARE TECHNOLOGY PROGRAM OF GERM, A MICROELECTRONICS AND COMPUTER TECHNOLOGY CORPORATION (MCC) SOFTWARE TOOL
An Informal Memorandum

Ernest M. Fridge III, Dennis Braley, Laura Pusch, and Charles Pitman
Software Technology Branch/FR5

ANTICIPATED NEED

We need a graphic user interface "standard." That is, we need a generic, easily reconfigured, graphic user interface to aid in the drawing and production of special graphs for input to/output from various types of computer programs, including FORTRAN Software Development and Maintenance Aids, Advanced Software Development Workstation (ASDW) software components, and Computer Aided Software Engineering (CASE) tools. Because of the growing popularity of both graphic and textual user interfaces, we believe that both types of interfaces must be provided for software tools in order to serve the entire user community.

SUMMARY EVALUATION OF GERM

GERM is a very nice and colorful, graphic user interface. Its schema files and Icon editor allow it to be custom-designed to produce various types of graphs. It appears to have a lot of potential as a graphic user interface. However, GERM does not currently include all of the capabilities required to produce some types of graphs needed by the Software Technology program, and it has some undesirable "features."

After weighing such considerations, we believe that GERM's potential as a generic, graphic user interface is large enough to justify a reasonable expenditure of funds, over the course of one or two years, so that MCC can provide GERM with the additional capabilities we need.

POSSIBLE APPLICATIONS FOR GERM

GERM might eventually be useable for the following projects. However, in each case, a translator would be needed to

1

transfer data back and forth between GERM files and the application's files.

1. FORTRAN Software Development and Maintenance Aids. GERM (the current version, except with minor enhancements to the mouse/keyboard interface) might be used as a graphical browsing front-end for the output data from the following processors (see attached appendix):
CCREF, DEPCHT, MAZE, RELREF, SETGEN, TDEM, and TOCGEN.

2. ASDW Software Components. GERM (the current version, except with minor enhancements to the mouse/keyboard interface) might be used as an Entity-Relationship (ER) diagram tool that would form the front-end browser/editor for Bauhaus, a software reuse tool being developed for NASA by Inference Corp.

3. CASE Tools. GERM (the current version) cannot be used as the front-end graphic interface for Data Flow Diagrams, IDEF-0 diagrams, nor other CASE diagrams. This is principally because GERM cannot currently enforce consistency between levels -1, 0, 1, 1.1, 1.1.1, etc., to the extent required by the various diagramming methodologies. This deficiency must be fixed by MCC, and probably can be during the recommended one to two year period for fixing all of GERM's deficiencies.

OTHER CRITICISMS

The following list of criticisms is not complete and is not intended to be construed as a complete list of our requirements for GERM modifications.

1. Pull-down menu selection of options is required as an alternative to the current mouse "chords." PFS Write is a good example of a program that has both mouse selection and menu selection.

2. A larger type font is required for the default type font. The current one is barely legible to many people.

3. The program is difficult to learn, probably because of the lack of true context-sensitive help. The Help facility seems more like a tutorial or a large on-line user's guide than a true context-sensitive help facility that can easily be used to diagnose your error.

2

4. The code needs to be made more portable by the use of an interface package such as TAE Plus (or X Windows directly) and by replacement of its Prolog code with C code.

Further comments by individual authors are attached.

COMMENTS ON THE GERM PROGRAM

Dennis Braley

1. My Intended use of GERM

From the beginning my intended use of GERM has been to use it as a graphical browsing front end for the output data from some of our Fortran analysis tools such as CCREF and RELREF. This use would be very similar to MCC's current use of the tool on c code in their DESIRE project.

2. Problems with GERM for my intended use

I feel that the user interface is not polished enough for my needs. It needs a much easier to use user interface such as that contained in PC programs such as:

1. Microsoft Windows

2. PFS Professional Write

The basic features needed are:

1. true context sensitive help

2. pull down menu selection of options as an alternative to its current mouse button sequences. PFS write is a good example of a program that has both key sequences and menu selection methods of execution.

3. The code needs to be made more portable by the use of an interface package such as TAE and by replacement of its Prolog code with c code.

The pull down menu method is needed as an optional method of running the program for those users who will be running GERM on an occasional basis. Unless a person is a full time user of a program, remembering complex mouse or key sequences is very difficult.

3. Conclusion

I do not think that NASA could justify the purpose of GERM just for my use because my current tools could be improved by simply adding a hyper text type browsing system front end. The graphical front end that GERM would supply might be desired by some users, but certainly could not be considered as required. Hyper text methods would certainly be adequate for most if not all user needs.

AN ANALYSIS OF GERM 3

BY

LAURA PUSCH

To: Ernie Fridge, Dennis Braley, Charles Pitman, Ken Crouse

<u>Our Needs</u> (using the RELREF tool as an example) can, I believe, be summarized as follows. We need a Graphical Software design System which will:

I.   Show relationships between parts and subparts while maintaining their modularity
II. Be flexible
III. Be capable of hiding information.
IV. Have a concise sorting capability
V.   Maintain uniformity of method throughout the graph design.
VI. Generate proper and easily understood documentation
VII. Have an easily understandable mousing capability
VIII. Provide pull-down window which are self-explanatory and an adequate help facility which is context-sensitive
IX. Code which is portable between UNIX and DOS (at a minimum)
X.   System and user file security.

GERM 3 (NOTES)

1.   Software appears to perform as was intended.

<u>Good points of the software</u>:

a.   <u>Graphical displays</u> good; however, horizontal <u>browser scrolling</u> is awkward-contains a "spring-back" characteristic which can waste a lot of time and confuse the new user.

b.   <u>Schema files</u> {in which the <u>icons, shapes,</u> etc. are described textually} can be custom-designed with the simultaneous use of the <u>Icon Editor,</u> which has a screen-paint canvas and is fairly easy to use. This enhances flexibility of design.

c.   <u>Graphical displays</u> perform as described in the <u>Schema file,</u> making simulation of other methodologies (like <u>SADT</u>) possibly in most instances.

d.   A data dictionary is maintained and can be scanned using <u>Index</u> and <u>edit</u>

<u>Problems with the software</u>:

a.   <u>Nesting of aggregates</u> does not appear to be feasible right now, above the first level. This could result in a very <u>large canvas</u> which would have to be scrolled over and over again detracting from the information hiding capability and ~~also from the~~ modularity. *This type of scrolling tends to wear the user out.*

b.   Program is difficult to <u>learn</u> and the mousing system is horrible

c.   the Unix software is too temperamental for a program like GERM:

   1.   it is constantly going down and appears to be "delicate" and easy to mess up
   Example: the inadvertent use of a common Dos command caused the entire ICON file to disappear with no backup; this necessitated reloading of the tape which, even now, is not properly installed.

2. <u>Security of the User Files</u> is not complete. Inadvertent file foul up could easily result.

d. <u>UNIX</u> is more difficult to learn than <u>Dos</u> and harder to maintain. Without UNIX experts available who also are familiar with GERM 3, the user spends a considerable amount of time troubleshooting instead of designing.

e. Mousing is most difficult to keep straight in GERM.

RECOMMENDATIONS TO NASA

1. Urge <u>MCC</u> to develop a <u>version</u> of GERM 3 which will run on either an <u>IBM PC</u> or a <u>MAC II</u>.
2. Appoint an <u>UNIX/IBIS3 expert</u> to assist users while GERM 3 is resident on the <u>SUN</u>.
3. Obtain a backup tape of the GERM 3 System and keep it in the software lab
4. Teach Lei Wang (or someone <u>else</u> who is also very knowledgeable about the SUN network how to properly install the software
5. Get a better editor than VI

RECOMMENDATIONS TO MCC

1. User interface for Schema Design
2. Nesting of Aggregates
3. Portability to MAC II or IBM PC
4. Improve horizontal scroll on canvas - eliminate "rubber band"
5. Simplify the mousing technique