DOE/ER/10521--T1

DOE/ER/10521--T1

DE82 009971

MASTER

INFORMATION DISPLAY SYSTEMS³⁴

Summary of Project Status

Report GWU-IIST-82-04

NOTICE

PORTIONS OF THIS REPORT ARE ILLEGIBLE.

It has been reproduced from the best available copy to permit the broadest possible availability.

Michael B. Feldman

James D. Foley

✓ March 1982

Institute for Information Science and Technology
Department of Electrical Engineering and Computer Science
School of Engineering and Applied Science

The George Washington University

Washington, D.C. 20052

This project is sponsored by the
U.S. Department of Energy
Grant DE-AS05-79ER10521

DISCLAIMER

This book was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

MBW

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency Thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

DISCLAIMER

Portions of this document may be illegible in electronic image products. Images are produced from the best available original document.

Tools for the Designers of User Interfaces

1. INTRODUCTION

In this overview paper we present our work toward developing methodologies and tools which can aid in the design and evaluation of user-computer interfaces. We want to impose structure on the typically very complex task of designing a user-computer interface, so the design can be divided into manageable pieces, each of which can be dealt with in a systematic, rigorous and at least partially quantitative way. We believe this will help make User Interface Design more of a science and less of an art, and lead to improved design.

The actual process of designing a user interface can be accomplished as four major steps, which we call the conceptual, semantic, syntactic, and lexical design steps. Each step can be dealt with in sequence; one after the other, with an occasional reexamination of a previous step. We call these four steps a design framework.

2. THE DESIGN FRAMEWORK

The conceptual design is the definition of the key application concepts which the user of the interface must understand in order to use the system. For a simple text editor, the key concepts are files; lines of a file, and operations (add, delete, move) on lines. The conceptual model, as in this case, typically defines objects, relations between objects (a line is

in a file), and operations on the objects, and sets the stage for the semantic design of the user-computer interface.

The semantic design deals with the functionality of the system to be accessed via the intermediary of the user interface. The user performs certain actions, calculations/processing ensues, and information is presented to the user. At the semantic design level we are concerned only with the meanings of the inputs, the processing, and the outputs; we are not concerned with the form or the sequence of the inputs and outputs.

The syntactic design deals with the sequence of the inputs and outputs. For the input, sequence is akin to grammar--the rules by which sequences of words in a language are formed into legitimate sentences. The types of words in an input sentence are typically commands, quantities, names, coordinates, or arbitrary text. As in English, the words are the units of meaning in the input and cannot be further decomposed without losing their meaning. For output, the notion of sequence is extended to include the spatial domain as well. Therefore the output syntax includes the 2D or 3D organization of a display as well as any temporal variation in the form. The "words" in the output sequence, by analogy to work in the input sequence, represent the units of meaning being conveyed from the computer to the user. The units of meaning are often conveyed graphically as symbols and drawings made up of lines, curves, and points rather than as words made up of letters.

The lexical design determines how words in the input and output are actually formed from the available hardware capabilities. For input, this involves designing the interaction techniques for the application. An interaction technique is a way of using a physical input device (tablet, keyboard, mouse, etc.) to input a certain type of word (command, value, coordinates, etc.). For example, some of the interaction techniques for command specification are, selecting from a menu with a light pen or with a cursor controlled by a mouse, typing of the command name on a keyboard, and speaking the name of the command into a speech recognizer.

For output, lexical design means combining the available hardware lexemes to form the symbols and shapes which are to be presented to the user, using the available hardware lexemes. For text output, this reduces to selecting text attributes such as font, size, color, and background color: and to spelling (i.e., combination of hardware lexemes, which form the character set) of words already defined in the dictionary. In other cases, such as situation displays, the symbols used must be designed and composed from lexemes such as lines and other graphics primitives, and the symbols must be assigned attributes such as color, intensity, linestyle, and size.

The nub of this four-level framework for design is found in formal language theory; the framework has been successively refined and reported in a series of papers [FOLE74, FOLE78, FOLE80, FOLE81b]. We have worked/are working with this framework

in several ways: the organization of design principles, the evaluation of existing user-computer interfaces, the evaluation of interaction techniques (which are the lexical-level design of the input), the formal specification of the syntactic and lexical design of input and output, the calculation of metrics of "goodness" based on the formal specification, and the design of an "abstract interaction handler" to remove much of the syntactic and lexical design from the application program.

3. ORGANIZING DESIGN PRINCIPLES

The past ten years have seen several user interface designers setting forth their design principles [BENN76, BRITT77, ENGE75, HANS71, WALL76] in the form of general objectives and specific do's and don't's. These papers plus personal experience form the knowledge base available to most designers. Often the criteria are soundly-based: a useful start in developing tools for designers is to organize the principles, showing how they apply at the conceptual, semantic, syntactic, and lexical design levels. This process has been partially completed, as reported in FOLE81b, for principles dealing with feedback, error correction, response time, consistency, and display structure.

4. EVALUATING USER-COMPUTER INTERFACES

Given an organized set of design criteria, it is possible to perform a systematic evaluation of existing user-computer interfaces by a combination of watching others use the interface and learning to use the interface oneself. In this process it is

critical to note idiosyncratic features of an interface when they are first encountered, lest one adjust to the features. Two such evaluations have been conducted: the first [HERB80] of DIDS, the Decision Information Display System used by the federal government for policy studies, the second [BLES81] of SEEDIS, the Socio-Economic Environmental Demographic Information System developed at Lawrence Berkeley Labs.

5. EVALUATION OF INTERACTION TECHNIQUES

Recall that an interaction technique is a way of using a physical input device to input a word, and hence is the lexical level input design.

Each type of word in the input language (commands, object, values, positions, etc.) corresponds to the type of interaction task which the user of the interactive system will have to carry out in the process of inputting a complete sentence. Hence each word corresponds to an interaction task. The designer of the interface, using knowledge of the task semantics, assigns values to each of the tasks requirements. For instance, one requirement of a command selection task is the size of the set of possible commands from which one command is to be selected. Similarly, one requirement of a value input task is the required resolutions of the value (one part in 100, one part in 4000, etc.)

Interaction techniques differ in their ability to meet task requirements: input of a value using a potentiometer can readily meet a resolution requirement of one part in 100, but can not

readily meet a one part in 10000 requirement, while input of a value by typing on a keyboard can easily meet any reasonable resolution requirement. Hence each technique has a set of attributes which describe how well it can meet the task requirements.

At the lexical design level, the designer chooses interaction techniques (and their associated devices) based in part on the ability of the techniques to meet the task requirements. In FOLE81a we (in conjunction with V. Wallace, in work also partially sponsored by the US Army Research Institute) have described and organized the interaction techniques by their associated task. We have enumerated the attributes of each technique, as well as the types of requirements each interaction task has. A number of germane human factors design issues have been identified for the techniques by drawing on the literature and the guidelines mentioned above. Nine experiments dealing with interaction techniques are also critically reviewed. A method of interaction technique diagrams is created, to aid in understanding, analyzing, and documenting the techniques and experiments. A diagram shows the cognitive, motor, and perceptual steps which the user of a technique performs. The report is meant as a guide to aid designers in selecting appropriate interaction techniques and devices.

6. FORMAL SPECIFICATION AND METRICS

The syntactic and lexical designs of a user interface should be describable by formal language tools, in the spirit (but not necessarily in the image) of BNF, regular expressions, and flow expressions. We are developing formal tools for describing both the input and output of a user interface, as well as the relationship between input and output. The input definition deals with concepts such as token types (which are the purposes of interaction techniques, as described above), sequences of tokens, and the binding of tokens to sequences of actions with physical devices. The output definition deals with concepts such as screen areas and their contents, and attributes (such as color, font, and linestyle) of tokens within various areas. Metrics treat issues such as complexity and consistency of syntactic rules, consistency in the use of codings, continuity of visual attention on the display, continuity of tactile motion with the interaction devices, and time required to input commands. The metrics draw upon the guidelines mentioned above.

The designer of a user interface will use the tools to describe the interface. This in itself helps create a more disciplined design environment. In addition, the formal definition will be processed, metrics evaluated, and potential design problems flagged for further attention by the designer. In the long run, the user interface definition will be input to an interaction handler which will actually implement the user interface.

Details on the work in formal specification and metrics are found in BLES82a and BLES82b.

7. MODULARITY IN INTERACTIVE SYSTEMS

If half the problem is a formal-specification problem, the other half is a problem in software construction. In particular, we are interested in determining the types of interactions and applications with the property that interaction-style-specific information can be separated from application semantics, so that a system "front end" can be built without disturbing the system itself, or rather that a system can be built without concern for interaction details so that these details can be easily and economically changed to suit different styles.

Being able to separate front-end development from system semantics is advantageous in a number of areas:

- o Controlled experimentation is facilitated since a number of different front ends can be built and tested experimentally with the assurance that the underlying system has been held constant.
- o Front ends can be tailored not just to different user classes but even to different individuals. This is particularly important for one type of user: the highly creative, highly-paid analyst whose morale and productivity might be greatly improved by the availability of an individualized interactive system.
- o Separation of application functions from interaction handling makes possible the development of libraries of interaction-handling modules with known human-factors attributes, thus relieving designers from analyzing each new system

separately. There is obviously an economic advantage as well whenever code can be re-used.

o Such separation also means that individuals can be trained as human-factors specialists (analogous to those in traditional ergonomics or industrial engineering) who need not be familiar with details of application systems. Similarly, writers of application functions need not be familiar with details of interaction handling.

o Portability of application systems is enhanced because interaction details are not embedded in functional (application) code.

In thinking about such modularity considerations, an analogy can be drawn to the DBMS approach to handling the "back end" of a system. There an application writer can request information to be drawn from a data base with little concern for its organization or access techniques. In the domain of interactive systems, the abstraction is done at the "front end" instead, where an application writer can request information from and display results to an interactive user without being concerned with the detailed interaction syntax or the device-level access methods.

To support this idea, we are designing what we have called an "abstract interaction handler (AIH)", roughly analogous to a DBMS, responsible for exchanging style-independent information with application modules. The AIH in turn communicates with a number of "style modules", which know the details of specific

interaction styles. These styles are roughly analogous to access methods. Binding of these interactions to specific devices will be handled by a standard graphics package such as the Core System.

Designing requirements for an AIH are discussed in FELD81a; Figure 1 gives a block diagram of the system design resulting from consideration of the requirements. This block diagram is serving as the overall framework for building our system.

A number of components of the system are currently in operation or nearing completion. These include an intermediate language and prototype interpreter [ROGE81, ROGE82a, ROGE82b, FELD82], a logical-screen handler [FELD81b] and a style module to handler "menu"-like interactions [KAMR82]. A library of programs supporting various interaction techniques is under development.

8. GRAPHICS RESEARCH

Our graphics work has been with a proposed standard device-independent graphics subroutine package: the Core System. We have implemented this proposed system with partial support from DOE. In implementing the Core, we have developed an internal implementation structure which is both flexible and efficient, and have thus contributed to a store of knowledge on appropriate implementation structures for such packages. The implementation strategy is described in WENN80 and FOLE81c, a detailed discussion of the issues surrounding the implementation of Core graphics input is found in HANL82.

Figure 1. MODULAR INTERACTIVE SYSTEM DIAGRAM

Our benchmarks have demonstrated that the cost of the general structure is no more than that of device-specific graphics packages. This has been a matter of debate in the graphics community. A paper describing the benchmarks is in preparation.

With DOE permission, we have begun to distribute the GW Core System implementation for a minimal fee which recovers the costs of doing the distribution. The Core is also distributed by DOE Argonne Code Center. Thus far the distribution is as follows:

DOE-related labs and universities - 6

Other government sites - 3

Other universities and non-profit groups - 8

Private sector - 13

Thus the DOE participation in Core System development has resulted in useful research in its own right, and a contemporary readily-available standard graphics package.

We have also addressed the question of suitability of the Core System for use with raster graphics equipment and for use with Information Display Systems in general. We have found the Core lacking in a number of areas, and are thus now developing a new conceptual model for use in developing graphics subroutine packages oriented toward raster graphics systems.

The fundamental problems are that the Core System recognizes neither the bit-map (pixel) representation of an image nor the desirability of the application programmer dealing directly with

pixel matrixes and them or logical/arithmetic combinations of them as output using primitives. Yet such operations are very useful, as testified to by single address space architectures such as PERQ, APOLLO, and STAR. Further, the operations cannot be subsumed in the present set of abstractions found in the Core System. Hence the need for a new model.

We expect this model to have impact on future directions in the development of graphics packages (which are used heavily in DOE work) for use in research projects and throughout the computer graphics community. The conceptual model for raster systems is reported in ACQU81. A detailed reference manual is in preparation, following which some portion of the extensions will be implemented as extensions to the current Core.

9. INTEGRATING THE MAJOR COMPONENTS

Figure 2 shows how the work on formal specification, metrics, the AIH, and the Core System will merge together. The user interface designer will specify the interface and evaluate it with the metrics. The designer may iterate several times at this stage. At any point the interface can actually be tried. Problems found while using the interface can easily be repaired, by modifying the interface specification, and several alternative interfaces can be tested and made available for regular use.

10. SYSTEM CONFIGURATION

Figure 3 lists the equipment installed as components of the VAX-based system we are running. Units whose acquisition was funded by this grant are indicated.

Figure 2. Integrated System Design

G.W.U. GRAPHICS LAB

COMPUTING ENVIRONMENT.

DEC VAX/11-780 with 2.5 Mbytes (VMS/EUNICE)

- Three 300-Mbyte disk drives, one tape drive
- Three Tektronix 401x graphics terminals
- One Ramtek 9400 Raster Graphics terminal
- One Matrix color camera
- One DMC-11 DECnet interface
- Two ADM3a terminals with Retrographics boards
- One Vector General VG3400 vector display
- DZ-11 terminal controller (8 ports)
- DZ-11 terminal controller (8 ports)
- assorted ASCII terminals ; 600 lpm printer

□ = funded under this research grant

Figure 3. Computer Configuration

REFERENCES

- ACQU81 Acquah, J., J. Foley, J. Sibert, and P. Wenner, "A Conceptual Model of Raster Graphics Systems," The George Washington University, Institute for Information Science and Technology Technical Report GWU-IIST-81-29, November 1981.
- BENN76 Bennett, J., "User-oriented Graphics Systems for Decision Support in Unstructured Tasks," Proceedings of ACM/SIGGRAPH Workshop on User-Oriented Design of Interactive Graphics Systems, Pittsburgh, PA., October 1976, pp. 3-11.
- BLES81 Bleser, T., P. Chan Mei Chu, "A Critique of the SEEDIS User Interface," The George Washington University, Institute for Information Science and Technology Technical Report GWU-IIST-81-04, March 1981.
- BLES82a Bleser, T., J. D. Foley, "Towards Specifying and Evaluating the Human Factors of User-Computer Interfaces," to appear in "Conference on Human Factors in Computer Systems", March 1982.
- BLES82b Bleser, T., "A Formal Specification and Computer-assisted Evaluation of Human-computer Interfaces. (M.S. thesis) The George Washington University, Institute for Information Science and Technology, Technical Report GWU-IIST-82-01, Washington, D.C., March 1982.
- BRIT77 Britton E., "A Methodology for the Ergonomic Design of Interactive Computer Graphics Systems, and Its Application to Crystallography," University of North Carolina at Chapel Hill, UNC Report No. TR-77-011, November 1977.
- ENGE75 Engel, S., and R. Granda, Guidelines for Man/Display Interfaces, IBM Poughkeepsie Laboratory, TR 00.2720, December 1975.
- FELD81a Feldman, M.B., and Rogers, G.W., "Design Requirements for Style-Independent Interactive Systems", The George Washington University, Institute for Information Science and Technology, Technical Report GWU-IIST-81-06, Washington, D.C., 1981.
- FELD81b Feldman, M.B., and Kamran, A., "A Logical-Screen Handler Package (SH 1) for the GWU Core Graphics System", The George Washington University, Institute for Information Science and Technology, Technical Report GWU-IIST-81-18, Washington, D.C.,
- FELD82 Feldman, M.B. and Rogers, G.T., "Toward the Design and Development of Style-Independent Interactive Systems," to appear in "Conference on Human Factors in Computer Systems", March 1982.

- FOLE74 Foley, J. and V. Wallace, "The Art of Natural Graphic Man-Machine Conversation," Proceedings IEEE 62(4), April 1974, pp. 462-470.
- FOLE78 Foley, J., "The Human Factors-Computer Graphics Interface," Proceedings of Symposium on Human Factors and Computer Sciences, Computer Systems Technical Interest Group, Human Factors Society, June 1978, pp. 103-114.
- FOLE80 Foley, J., "The Structure of Command Languages," in R.A. Guedj, et al., eds., Methodology of Interaction, North-Holland, Amsterdam, 1980, pp. 227-234.
- FOLE81a Foley, J., V. Wallace, and P. Chan, "The Human Factors of Interaction Techniques," The George Washington University, Institute for Information Science and Technology, Technical Report GWU-IIST-81-03, Washington D.C., March 1981.
- FOLE81b Foley, J., "A Methodology for the Design and Evaluation of User Computer Interfaces," The George Washington University, Institute for Information Science and Technology, Technical Report GWU-IIST-81-05, Washington, D.C., March 1981.
- FOLE81c Foley, J. and P. Wenner, "The George Washington University Core System Implementation", proceedings SIGGRAPH '81, published as Computer Graphics 15(3), August 1981, pp. 123-132. Also GWU Technical Report GWU-IIST-80-12, 1980.
- HANL82 Hanlon, J., "Implementation and Evaluation of Input Functions for the 1979 Core Graphics System," The George Washington University, Institute for Information Science and Technology, Technical Report GWU-IIST-82-03, Washington, D.C., 1982.
- HANS71 Hanson, W., "User Engineering Principles for Interactive Systems," Proceedings 1971 Fall Joint Computer Conference, pp. 523-532.
- HERB80 Herbert, T., "Evaluation of the User-Computer Interface Design of the Domestic Information Display System," The George Washington University, Department of Electrical Engineering and Computer Science, Technical Report GWU-EECS-80-07, Washington, D.C., 1980.
- KAMR82 Kamran, A., M.B. Feldman, and J.D. Foley, "Interaction Handler System Design Document, The George Washington University, Institute for Information Science and Technology, Technical Report GWU-IIST-82-07, Washington, D.C., 1982.

- ROGE81 Rogers, G.T., and M.B. Feldman, "An Intermediate Language and Interpreter for Style-Independent Interactive Systems," The George Washington University, Institute for Information Science and Technology, Technical Report GWU-IIST-81-21, Washington, D.C., 1981.
- ROGE82a Rogers, G.T., "Interaction Handler System Overview and Structure", The George Washington University, Institute for Information Science and Technology, Technical Report GWU-IIST-82-02, March 1982.
- ROGE82b Rogers, G.T., "Structuring Interactive Systems for Flexibility of the User Interface (M.S. Thesis) March 1982, GWU-IIST-82-07.
- WENN80 Wenner, P., et. al., "Design Documentation of the GWU Implementation of the 1979 GSPC Core System", The George Washington University, Institute for Information Science and Technology, Technical Report GWU-IIST-80-06, 1980.
- WALL76 Wallace, V., Summary of "Conversational Ergonomics", Session ACM/SIGGRAPH Workshop on User-Oriented Design of Interactive Graphics Systems, Pittsburgh, PA., October 1976, pp. 121-122.

FINANCIAL STATEMENT FOR DE-AS05-79ER10521

AS OF JANUARY 31, 1982 (Rounded off)

ACCT DESCRIPTION	1st <u>AWARD</u> 9/1/79- 8/31/80	2nd <u>AWARD</u> 9/1/80- 8/31/82	BUDGET CHANGES	ALLOTMENTS AS OF 1/31/82	EXPENDITURES AS OF 1/31/82	CURRENT ENCUMBRANCES AS OF 1/31/82	BALANCE AS OF 1/31/82
SALARIES	79,570	171,018	(1,690)	248,898	217,195		31,703
WAGES	-0-	-0-	2,351	2,351	2,281		70
CONSULTANTS	-0-	8,400		8,400			8,400
LEAVE CREDITS	-0-	-0-	(2,110)	(2,110)	(2,110)		
COMMUNICATIONS	1,000	2,300	(657)	2,643	1,724	445	474
TRAV CONSULTANTS	-0-	-0-	250	250	238		12
TRAVEL DOMESTIC	4,000	10,000	750	14,750	10,491	450	3,809
PUB & PRINT	1,474	2,300	(1,443)	2,331	31		2,300
TRAINING EXP	-0-	-0-	700	700	700		
CONF EXPENSE	-0-	1,000		1,000			1,000
SERVICES	1,000	-0-	5,269	6,269	5,429		840
SUPPLIES	1,000	6,000	482	7,482	3,671	220	3,591
BOOKS	-0-	-0-	941	941	796	41	104
COMPUTER USAGE	14,000	32,300	(5,139)	41,161	31,599		9,562
RENTAL (S/W License)	-0-	-0-	300	300			300
FRINGE BENEFITS	13,129	40,189	(10,820)	42,498	39,962		2,536
INDIRECT COSTS	54,108	131,284	440	185,832	147,129		38,703
RESERVE	-0-		8,176	8,176			8,176
COST SHARING	(3,131)	(4,587)		(7,718)	(5,943)		(1,775)
EQUIPMENT	<u>143,850</u>	<u>53,911</u>	2,200	<u>199,961</u>	<u>157,598</u>	<u>20,473</u>	<u>21,890</u>
TOTALS	310,000	454,115		764,115	610,791	21,629	131,695

FINAL INCREMENT OF
THIRD-YEAR BUDGET ESTIMATE
JUNE 1, 1981 - AUGUST 31, 1982

A. Salary and Wages

Feldman (100% for 2 summer months)	6,930
Foley (100% for 2 summer months)	8,775
Research Associate (100% for 3 months)	4,640
Secretary (50% for 3 months)	1,620
4 Graduate Research Assistants (100% for 3 months)	<u>16,000</u>
TOTAL SALARIES	37,965

B. Fringe Benefits (17.6% of A)* 6,682

C. Computer Usage 10,000

D. Publications & Printing 2,000

E. Communications & Shipping 1,200

TOTAL DIRECT COSTS 57,847

F. Indirect Costs (50% of MTDC)** 28,293

Total Estimated Costs 86,770

Cost Sharing at 1% (868)

PROJECTED GRANT REQUESTED 85,902

*Fringe Benefits Rate fixed through 6/30/82 and provisional thereafter at 17.6% of salaries by negotiation agreement with DHHS 4/6/81.

**Indirect Cost Rates fixed through 6/30/83 and provisional thereafter at 50% of Modified Total Costs by negotiation agreement with DHHS 4/6/81.

INFORMATION DISPLAY SYSTEMS

List of Publications - March 1982

- GWU-EECS-80-06 Wenner, P. et al.
Design Document for the GWU Implementation of the
1979 GSPC Core System.
- GWU-EECS-80-07 Herbert, T.
Evaluation of the User-Computer Interface Design
of the Domestic Information Display System.
June 1980.
- GWU-EECS-80-12 Foley, J.D. and Wenner, P.
The George Washington University Core System
Implementation. Presented at 1981 SIGGRAPH
Conference, published as Computer Graphics 15(3),
August 1981, pp. 123-132.
- GWU-IIST-81-01 Garrett, M. and Foley, J.
Graphics Programming Using Data and Dependency
Declarations.
- GWU-IIST-81-02 Feldman, M.B.
Information Hiding in Pascal: Packages and Pointers.
April 1981.
- GWU-IIST-81-03 Foley, J.D., Wallace, V.L., and Chan, P.
The Human Factors of Graphic Interaction
Tasks and Techniques.
January 1981.
- GWU-IIST-81-04 Bleser, T., and Chan, P.
A Critique of the SEEDIS User Interface.
March 1981.
- GWU-IIST-81-05 Foley, J.D.
A Methodology for the Design and Evaluation of
User-Computer Interfaces.
March 1981.
- GWU-IIST-81-06 Feldman, M.B. and Rogers, G.T.
Design Requirements for Style-Independent
Interactive Systems.
June 1981.

- GWU-IIST-81-07 Foley, J.D.
Tools for the Designers of User Interfaces.
Presented at Workshop/Symposium on Human
Computer Interaction, Georgia Institute of
Technology. Published in revised form as
"The Design and Evaluation of User Interfaces",
Proceedings 1981 National Computer Graphics Association
March 1981.
- GWU-IIST-81-17 Feldman, M.B., Kamran, A., and Rogers, G.T.
Information Display Systems Project Interaction
Handler Development Summary of Modules.
October 1981.
- GWU-IIST-81-18 Feldman, M.B. and Kamran, A.
A Logical Screen-Handler Package (Sh 1) for
the GWU Core Graphics System.
August 1981.
- GWU-IIST-81-19-2 Feldman, M.B. and Foley, J.D.
Information Display Systems Project Index
of Technical Reports.
October 1981.
- GWU-IIST-81-20 Holmes, H., Thomas, J., and Rogers, G.T.
A Clause-structured Interaction Handler.
(In Process)
- GWU-IIST-81-21 Rogers, G.T. and Feldman, M.B.
An Intermediate Language and Interpreter for
Style-Independent Interactive Systems.
August 1981.
- GWU-IIST-81-22 Bleser, T., and Foley, J.D.
Towards Specifying and Evaluating the Human Factors
of User-computer Interfaces.
August 1981.
- GWU-IIST-81-23 Feldman, M.B.
Tools to Facilitate Human-Factors Improvement in
Interactive Information Display Systems.
Presented at IEEE Fall COMPCON, Washington, D.C.
September 1981.
- GWU-IIST-81-24 Feldman, M.B.
An Application-Driven Model for an Abstract
Interaction Handler.
(In process)

GWU-IIST-81-29 Acquah, J. et. al.
A Conceptual Model of Raster Graphics Systems
November 1981.

GWU-IIST-82-01 Bleser, T.
A Formal Specification and Computer-assisted
Evaluation of Human-computer Interfaces (M.S. Thesis)
March 1982.

GWU-IIST-82-02 Rogers, G.T.
Interaction Handler System Overview
and Structure.
March 1982.

GWU-IIST-82-03 Hanlon, J.
Implementation and Evaluation of Input
Functions for the 1979 Core Graphics System.
(M.S. Thesis)
1982.

GWU-IIST-82-06 Rogers, G.T.
Structuring Interactive Systems for Flexibility
of the User Interface (M.S. Thesis).
March 1982.

GWU-IIST-82-07 Kamran, A., Feldman, M.B., and Foley, J.D.
Interaction Handler System Design Document.
1982.

GWU-IIST-82-08 Bleser, T., and Foley, J.D.
Towards Specifying and Evaluating the Human
Factors of User-computer Interfaces.
To appear in "Conference on Human Factors in
Computer Systems".
March 1982.

GWU-IIST-82-09 Feldman, M.B., and Rogers, G.T.
Toward the Design and Development of
Style-independent Interactive Systems.
To appear in "Conference on Human Factors in
Computer Systems".
March 1982.