Transportable Applications Environment (TAE) Tenth Users' Conference

Edited by
Chris Rouff
NASA Goddard Space Flight Center
Greenbelt, Maryland

and

Elfrieda Harris
Arleen Yeager
RMS Technologies, Inc.
Lanham, Maryland

Presentations from a conference sponsored by the
NASA Goddard Space Flight Center
and held at the Goddard Space Flight Center
Greenbelt, Maryland
June 14–17, 1993
PREFACE

Goddard Space Flight Center sponsored the Tenth TAE Users' Conference on June 14-17, 1993 held at Goddard.

This document represents the proceedings of the Tenth TAE Users' Conference. The presentations included in these proceedings were published as received from the authors with little modification and editing.
Message From the TAE Project Manager

This was the first TAE conference that followed an informal workshop format with panel discussions, demonstrations, tutorials and working sessions. It provided a chance for all attendees to participate - and they did! The informal format worked extremely well and helped to create a very interactive environment. Attendees actively participated, and there was a good exchange of information and experiences between TAE users and developers. This feedback from many of you will help us plan future directions for TAE Plus.

The Tenth TAE Conference is the last TAE Users’ Conference that Goddard Space Flight Center will coordinate. With the software being transferred into the private sector, all future user conferences will be managed by Century Computing, Inc., the commercial developers/distributors of TAE Plus. On this note, the conference offered a great opportunity for the TAE Project Office, the TAE Support Office and Century Computing to respond to TAE users’ questions, concerns and comments about the commercialization of TAE. Several of the presentations discussed more details about the transfer and described what will be available in TAE Version 5.3, the first commercial release. I think we all came away with a better understanding of what the technology transfer “means to me”.

Many thanks to each and everyone who participated in the conference.

Chris Rouff
TAE Project Manager
NASA/Goddard

Acknowledgements

The TAE Project would like to express its appreciation to everyone who demonstrated their application at the conference and to all those who participated in the panel sessions. In addition, we would like to thank the following individuals for their significant roles in planning and organizing the conference:

Elfrieda Harris, TAE Support Office, RMS Technologies, Inc.
Arleen Yeager, TAE Support Office, RMS Technologies, Inc.

TAE is a NASA software project within the Data Systems Technology Division at Goddard Space Flight Center with contract support by Century Computing, Inc. The work is sponsored by NASA’s Office of Space Communications.
Presentations from the
Tenth TAE Users' Conference
June 14-17, 1993

Sponsored by
Goddard Space Flight Center

Held at
Goddard Space Flight Center
Greenbelt, Maryland

Table of Contents

User Experiences with C++ .............................................................. 1
David Fout, Century Computing Inc. .................................................. 3
Elizabeth Wei, Siemens Corp. Research ............................................. 13

User Experiences with Ada .............................................................. 21
Christina L. Langford, Coastal Systems Station ............................... 23
Roger Sheldon, Loral AeroSys ........................................................ 35

Keynote Address ............................................................................... 45
Managing the Design of the User Interface
Deborah Mayhew, Deborah Mayhew and Associates ....................... 47

Technology Transfer ......................................................................... 75
TAE Yesterday, Today & Tomorrow
Marti Szczur, Goddard Space Flight Center ..................................... 77

TAE V5.3 Summary ........................................................................... 85
The Commercialization of TAE Plus
Don Link, Century Computing Inc. .................................................. 87

Usability and Application Testing ..................................................... 105
Computer-Human Interaction Models (CHIMES)
Jianping Jiang, CTA Inc. .................................................................. 107
User Interaction Testing of the User Interface
Jim Hicinbothom, CHI Systems Inc. ............................................... 115
User Interface Design Guidelines
Susan Adams, Pacific Northwest Lab .............................................. 123
TAE Plus V5.3 Testing Tools
Phil Miller, Century Computing Inc. ............................................... 131

Data Driven Objects ........................................................................ 137
TAE Plus V5.3 Extensions to DDOs
Karl Wolf, Century Computing Inc. ............................................... 139
Data Driven Objects Potential Enhancements
Terry Bleser, Century Computing Inc. ............................................. 153
User Experiences with Data-Driven Objects
Patricia M. Jones, University of Illinois .......................................... 163
Object Dependencies ................................................................. 167

Item-to-Item Connections
  Craig Warsaw, Century Computing Inc. ................................. 169

Using the Spreadsheet Model of Computation for Defining Object Dependencies
  Margi Klemp, University of Colorado ................................. 177

Integration with Other Software ............................................. 187

The Development of a Graphical User Interface to the fault
Isolation System Database Manager
  Christopher Barclay, Joseph Molnar, Naval Research Lab .......... 189

UIL Support and Mrm Code Generation
  Kenneth Sall, Century Computing Inc. .................................. 197

CenterLine's Object Center C++ Compiler
  Greg Shirah, Goddard Space Flight Center .......................... 215

List of Conference Attendees ................................................ 223
User Experiences with C++

David Fout
Century Computing Inc.

Elizabeth Wei
Siemens Corp. Research
Tenth TAE Users' Conference '93

TAE Plus v5.2 User's Experiences with C++

David Fout
Century Computing, Inc.
1014 West Street
Laurel, MD 20707
(301) 953-3330
Internet: dfout@cen.com

June 14, 1993
Outline

- Using the GNU g++ compiler
- Using the ObjectCenter Environment
- Data Manipulation in a TaePanel constructor
- TaePanelFile
- Examples of TAE and C++
• On Sun platforms, TAE Plus v5.2 is delivered with libraries built with the Sun C++ 2.0 C++ compiler.

• However, it also tested with g++ 1.40.3 on a Sun. If you want to use g++, you must recompile the entire tree. See Building TAE Plus from Source. (g++ can't link with Sun C++ compiled object code.)
Using the ObjectCenter Environment

- Due to a bug in ObjectCenter 1.2, many items will not appear in the panels when running a debug session. (You can get a tedious workaround from the TAE Support Office if necessary.)
- This bug was fixed in ObjectCenter 2.0.
Data Manipulation in a TaePanel Constructor

- When dynamically changing information about a panel or its items in the panel's constructor, you must use the TaeVar or TaeVarTable classes.

- The TaePanel and TaeItem class can't be used because the Wpt panels have not been created yet. They are created by the TaePanel::Show method.

```cpp
panel1C::panel1C (TaeCollection *collect) : TaePanel ("panel1", collect)
{
    // create an instance of each item in the panel.
    new TaeItem (this, "button1", &button1_React);
    TaeVarTable* viewTable = this->ViewTable();
    TaeVar* panelVar = viewTable->GetTaeVar("_panel");
    TaeVar* itemVar = viewTable->GetTaeVar("button1");
    itemVar->Set("fg","black");
    panelVar->Set("bg","white");
}
```
TaePanelFile

- TaePanelFile objects allow the user to register and field event sources such as file and socket descriptors.
- C++ binding to Wpt_AddEvent.
  - Subclass to give virtual methods your own functionality.

```cpp
class PanelFileC : public TaePanelFile
{
  public:
    PanelFileC (int);
  protected:
    virtual int HandleEvent (const TaeEventHandler&);
};
```
int PanelFileC::HandleEvent(const TaeEventHandler& )
{
    #define BUFFER_SIZE 132
    #define MIN_BYTE 1
    
    char buffer[BUFFER_SIZE];
    int n;
    // NOTE: There are a few extra file events that can and are
    // being ignored
    int fd = this->Descriptor();
    n = read(fd, buffer, BUFFER_SIZE-1);
    while (n>=MIN_BYTE)
    {
        printf("data byte = %s \t returned block size = %d byte
\n", buffer, n);
        n = read(fd, buffer, BUFFER_SIZE);
    }
    return 0;
}
TaePanelFile (cont)

• Create an instance
  
  PanelFileC* PanelFile = new PanelFileC(fd);

• Now you must register the instance with the TaeEventHandler. One option is to register the instance in the subclass constructor.
  
  PanelFileC::PanelFileC (int d) : TaePanelFile (d)
  {
    eventHandler->Register(this, (int) WPT_UPDATE_READMASK);
  }

• De-register the object with TaeEventHandler. This is often done in the subclass' destructor.
  
  eventHandler->Deregister(PanelFile);
Examples of TAE and C++

- `$TAEDEMONSRC/ddodemo.cc` and `$TAEDEMONSRC/timerdemo.cc` are two C++ programming examples delivered with TAE v5.2.
- New v5.2 Tips and Tricks document (coming soon)
Object (Tae Item) Cloning

Instead of creating some maximum number of items on the panel, create only a sample in the workbench. At run time, based on certain information at hand, dynamically create needed new items by making each a clone of the sample item.

Rationale

- Number of items is unknown until run time.
- One event handler for all items.
Sample Objects Created using the Workbench

Class Name: ???

Sort   List

☐   ☐ name (type)
# Objects Created At Run Time Through Cloning

<table>
<thead>
<tr>
<th>Class Name: IMAGE</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Sort</td>
<td>List</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>□</td>
<td>□</td>
<td>CreateDate</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>□</td>
<td>□</td>
<td>Description</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>□</td>
<td>□</td>
<td>Owner</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>□</td>
<td>□</td>
<td>Color</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>□</td>
<td>□</td>
<td>Format</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>□</td>
<td>□</td>
<td>Columns</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>□</td>
<td>□</td>
<td>Rows</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>□</td>
<td>□</td>
<td>IconID</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>□</td>
<td>□</td>
<td>BWIcon</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Problems Encountered

Protected clone method(s) (5.2 beta)

Work-around/Solution

My own 'clone' function

Much Better Solution

TAE+ 5.2
Work-around/Solution: my own 'clone' function

1. instantiate a new item:

   new_item = new sample_item_class (panel,new_item_name)

2. furnish the new item:

   - extract from the sample the resource values (both common to all presentation types and specific to the type being dealt with)
   - set these resources for the newly instantiated item with extracted values except for a new location (i.e., the 'origin')
The Better Solution: official TAE+5.2

1. instantiate a new item:

   new_item = new sample_item_class(panel,new_item_name,
   
   \&react_func)

2. furnishing:

   View/TargetTable()\rightarrow Add(sample_item\rightarrow GetView/TargetVar()\n
   \rightarrow Clone(new_item_name))
User Experiences with Ada

Christina L. Langford
Coastal Systems Station

Roger Sheldon
Loral AeroSys
COMBAT SYSTEM TRAINING SIMULATOR

OPERATOR-MACHINE INTERFACE DEVELOPMENT

Using TAE+ and Ada

Christina L. Langford
Coastal Systems Station
Panama City Beach, FL

email: langford@phoebus.ncsc.navy.mil
SYSTEM OVERVIEW

Combat System Training Simulator:
Shipboard simulator to provide training for different combat systems.

Operator-Machine Interface (OMI):
Enables person on-board ship to
- Build scenario files for exercise
- Initiate a simulation exercise
- Monitor trainee performance during exercise
- Perform database management functions
DEVELOPMENT ENVIRONMENT

- Silicon Graphics 4D/440VGX
- IRIX 4.0.5
- NCD 19" X-Terminal
- Verdix Ada Compiler
- TAE+ v5.1
  resupgrade to v5.2
- STARS Ada/X Bindings
TAE+ With STARS Ada/X Bindings

- TAE+ libraries contain "X_Windows"
- TAE code generator generates "X_Windows"
- STARS Ada/X Bindings contain "X_Lib"

**Modification of TAE+ libraries**
- replaced "X_Windows" with "X_Lib"

**Modification of TAE generated code**
- replaced occurrences of "X_Windows" with "X_Lib"
OMI MAIN
MODIFICATION TO EVENT_LOOP

If SW error,

Display Error Panel;
-- Selections are "IN_PROGRESS" or "NOT_IN_PROGRESS"

Set Selection to "NOT_IN_PROGRESS";

Else if Selection is "NOT_IN_PROGRESS"

If an event is pending (Wpt_Pending)

When Event_Type is WPT_PARM_EVENT =>

Set up User_Context_Ptr;
Set Event_Code;
Set Selection to "IN_PROGRESS";

Endif

Else if Selection is "IN_PROGRESS"

Dispatch_Panel;
Set Selection to "NOT_IN_PROGRESS";

Endif
EXERCISE SITUATION DISPLAY UPDATES

Update Dynamic Text Fields
  Wpt_SetIntg, Wpt_SetReal, Wpt_SetString

Update Status Message Area
  message_vec : TAE.s_vector(1..n);
  Vm_SetString_vec(Info.View,"item", n, message_vec, P_UPDATE);
  Wpt_ViewUpdate(Info.Panel_Id, "item", Info.View, "item");

Update Label and Color of Pushbutton
  Vm_SetString(Info.View, "item", 1, "newlabel",P_UPDATE);
  Vm_SetString(Info.View, "item.bg", 1, "color", P_UPDATE);
  Wpt_ViewUpdate(Info.Panel_Id, "item", Info.View, "item");
Extracting Multiple Selections from Selection List

ptr_to_selections : variable_ptr
sel_count : taeint
selections : array (1..1) of (1..tae_taeconf.STRINGSIZE)
my_list : array (1..n) of string (1..s)
my_count : integer

Vm_Find(Info.target, "item", ptr_to_selections)
Vm_Extract_Count(ptr_to_selections, sel_count)
my_count := integer(sel_count)
For J in 1..sel_count loop
   Vm_Extract_SVAL(ptr_to_selections, J, selections(J))
   my_list(integer(J)) := selections(J)(1..s)
end loop
EXERCISE SITUATION DISPLAY UPDATES (cont.)

Retrieving User Selected Position in X Workspace

-- This code is in X Workspace event handler

X_Event : X_Lib.Events.Event
X_Window_Id : X_Lib.Window
X_Window_X, X_Window_Y : X_Lib.Coordinate

Wpt_ItemWindow(Info.Panel_Id, "xworkspace", X_Window_Id);
Wpt_Extract_Parm_xEvent(Global.Event_Ptr, X_Event)

When X_Event.Kind is Button_Press or Button_Release =>
XQueryPointer(...,X_Window_Id,...,X_Window_X, X_Window_Y,..)

-- X_Window_X and X_Window_Y contain the user
-- selected position in the X Workspace.
CONCLUSION

TAE PROVIDES RAPID DEVELOPMENT AND EASY MAINTENANCE OF USER INTERFACE

TAESO AND BULLETIN BOARD ARE ALWAYS HELPFUL

TAE ON-LINE MAN PAGES (V5.2) LACK ADA SPECIFIC LIBRARY ROUTINES

NOT ALL TAE LIBRARY ROUTINES WORK CORRECTLY
TAE+ in Ada

Using Ada with TAE+

Roger Sheldon

Loral AeroSys
Agenda

- Describe application
- Discuss Pros and Cons of using TAE+ and Ada
- Summary
Overview of Application

- Developed a planning and scheduling tool, SORTIM, for the US Air Force.
- Performs resource scheduling for student pilot training. Resources include students, instructor pilots, aircraft, simulators, and classrooms.
- SORTIM is based on ROSE, the Request Oriented Scheduling Engine. ROSE was developed by Loral AeroSys for NASA Goddard Space Flight Center. ROSE has it's own GUI developed in Ada using TAE+, Motif, and X Windows.

Loral AeroSys
Overview of Application, cont.

- SORTIM is based on ROSE, but has a completely different user interface, also developed using TAE+, Motif, and X Windows.
Pros of Using TAE+ and Ada

- By using Ada, the programmer is less likely to write buggy code due to Ada's strong type checking.
Cons of Using TAE+ and Ada

- TAE+ was written in C++. The Ada bindings to TAE+ do not match exactly with the programmer’s interface available to the C++ programmer.
- The underlying Ada bindings to Motif and X Windows also fail to provide all the features available to the C programmer.
- Some of the Ada bindings are broken.
Summary

- Using TAE+ to develop the SORTIM GUI saved considerable time.
- Given a choice, the best language to use with TAE+ is C++.
Keynote Address

Managing the Design of the User Interface

Deborah Mayhew
Deborah Mayhew and Associates
Managing the Design of the User Interface

Prepared for

TAE
Tenth Users' Conference
June 1993

Prepared by

Deborah J. Mayhew & Associates
Panhandle Road
PO Box 248
West Tisbury, Massachusetts 02575-0248
508-693-7149
Managing the Design of the User Interface

AGENDA

• What Makes an Interface Usable?
• How Do You Do It?
• Why Should You Care?
• Who Else is Doing It?
• What Has Their Experience Been?
• How Can You Get Started?

WHAT MAKES AN INTERFACE USABLE?

• Design Issues
• Example One: Screen Design
• Example Two: Organization of Functionality
• Example Three: Color
• Example Four: I/O
Managing the Design of the User Interface

WHAT MAKES AN INTERFACE USABLE?

Design Issues

User Model

Dialog Style

I/O Devices

Organization of Functionality

Screen Layout and Design

Response Time

Error Handling

User Documentation

WHAT MAKES AN INTERFACE USABLE?

Example ONE: Screen Design

POOR:
WHAT MAKES AN INTERFACE USABLE?
Example ONE: Screen Design

PRINCIPLES:
- Poor use of white space
- No grouping
- Group titles not distinguished from captions - hard to find
- Horizontal orientation of menus - hard to read
- Poor ordering
- Inconsistent button location

IMPROVED:
Managing the Design of the User Interface

**WHAT MAKES AN INTERFACE USABLE?**

Example TWO: Organization of Functionality

**POOR:**

<table>
<thead>
<tr>
<th>Item</th>
<th>Description</th>
<th>Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Gloria Vanderbilt Jeans</td>
<td>$125.00</td>
</tr>
<tr>
<td>2.</td>
<td>Gloria Vanderbilt Jeans</td>
<td>$125.00</td>
</tr>
<tr>
<td>3.</td>
<td>Gloria Vanderbilt Jeans</td>
<td>$125.00</td>
</tr>
<tr>
<td>4.</td>
<td>Mohair Turtleneck Sweater</td>
<td>$210.00</td>
</tr>
<tr>
<td>5.</td>
<td>Mohair Turtleneck Sweater</td>
<td>$210.00</td>
</tr>
<tr>
<td>6.</td>
<td>Mohair Turtleneck Sweater</td>
<td>$210.00</td>
</tr>
<tr>
<td>7.</td>
<td>Reversible Disco Bag</td>
<td>$55.00</td>
</tr>
<tr>
<td>8.</td>
<td>Italian Sling Back Pumps</td>
<td>$175.00</td>
</tr>
<tr>
<td>9.</td>
<td>Italian Sling Back Pumps</td>
<td>$175.00</td>
</tr>
</tbody>
</table>

To View Item Summary, Press Item Number and ENTER
To View Next Page, Press NEXT
To Leave Form, Press CANCEL

---

**WHAT MAKES AN INTERFACE USABLE?**

Example TWO: Organization of Functionality

**POOR:**

<table>
<thead>
<tr>
<th>Item</th>
<th>Description</th>
<th>Size</th>
<th>Color</th>
<th>Quantity</th>
<th>Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.</td>
<td>Mohair Turtleneck Sweater</td>
<td>L</td>
<td>Jade</td>
<td>1</td>
<td>$210.00</td>
</tr>
</tbody>
</table>

To Change Item, Press ENTER
To View Next Item Summary, Press NEXT
To View Previous Item Summary, Press PREV
To Return to Order Form, Press CANCEL
WHAT MAKES AN INTERFACE USABLE?
Example TWO: Organization of Functionality

PRINCIPLES:

Overtaxes human short-term memory

Tedious navigation

Organization does not support user task

IMPROVED:

Company ABC
ORDER FORM

<table>
<thead>
<tr>
<th>NO.</th>
<th>ITEM</th>
<th>COLOR</th>
<th>SIZE</th>
<th>PRICE</th>
<th>QUANTITY</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Jeans</td>
<td>Black</td>
<td>S</td>
<td>$125.00</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>Jeans</td>
<td>Black</td>
<td>M</td>
<td>$125.00</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>Jeans</td>
<td>Black</td>
<td>L</td>
<td>$125.00</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>Sweater</td>
<td>Red</td>
<td>S</td>
<td>$210.00</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>Sweater</td>
<td>Blue</td>
<td>S</td>
<td>$210.00</td>
<td>1</td>
</tr>
<tr>
<td>6</td>
<td>Sweater</td>
<td>Orange</td>
<td>S</td>
<td>$210.00</td>
<td>1</td>
</tr>
<tr>
<td>7</td>
<td>Dance Bag</td>
<td></td>
<td></td>
<td>$55.00</td>
<td>1</td>
</tr>
<tr>
<td>8</td>
<td>Pumps</td>
<td>Black</td>
<td>6.5M</td>
<td>$175.00</td>
<td>1</td>
</tr>
<tr>
<td>9</td>
<td>Pumps</td>
<td>Black</td>
<td>9M</td>
<td>$175.00</td>
<td>1</td>
</tr>
</tbody>
</table>

To View Item Summary, Press Item Number and ENTER
To View Next Page, Press NEXT
To Leave Form, Press CANCEL
WHAT MAKES AN INTERFACE USABLE?

Example THREE: Color

PRINCIPLES:

Use color sparingly
Use color to support the user's task
Use color consistently
Provide good foreground/background contrast
Color contrasts better with black or white than with other colors
Use lighter (vs. darker) colors to draw attention
Avoid saturated blue for text
Exploit cultural color associations
"Warm" colors appear larger than "cool" colors
Allow users to turn color coding off or ask for different coding criteria
Managing the Design of the User Interface

WHAT MAKES AN INTERFACE USABLE?
Example THREE: Color

IMPROVED:

<table>
<thead>
<tr>
<th>Date</th>
<th>System</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>4/13/93</td>
<td>XYZ SYSTEM</td>
<td>3:30 pm</td>
</tr>
</tbody>
</table>

Accounts Payable

<table>
<thead>
<tr>
<th>Name</th>
<th>Account #</th>
<th>DUE Date</th>
<th>Paid</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alberts, S.</td>
<td>123-45</td>
<td>4/1/93</td>
<td>Yes</td>
</tr>
<tr>
<td>Canan, G.</td>
<td>416-44</td>
<td>4/1/93</td>
<td>No</td>
</tr>
<tr>
<td>Fisher, G.</td>
<td>334-01</td>
<td>4/15/93</td>
<td>No</td>
</tr>
<tr>
<td>James, R.</td>
<td>214-91</td>
<td>4/28/93</td>
<td>No</td>
</tr>
<tr>
<td>Jones, F.</td>
<td>967-23</td>
<td>4/7/93</td>
<td>Yes</td>
</tr>
<tr>
<td>March, K.</td>
<td>441-88</td>
<td>4/12/93</td>
<td>No</td>
</tr>
</tbody>
</table>

To scroll forward, Press DOWN  To exit, Press CANCEL
To scroll back, Press UP

WHAT MAKES AN INTERFACE USABLE?
Example FOUR: I/O

SOUND AS OUTPUT

"One can imagine how a single sound could be used to give information about a file arriving in a message system. The file hits the mailbox, causing it to emit a characteristic sound. Because it is a large message, it makes a rather weighty sound. The crackle of paper indicates a text file - if it had been a compiled program, it would have clanged like metal. The sound comes from the left and is muffled: The mailbox must be in the window behind the one that is currently on the left side of the screen. And the echoes sound like a large empty room, so the load on the system must be fairly low. All this information from one sound!"

Managing the Design of the User Interface

HOW DO YOU DO IT?

- Principles and Guidelines
- Methods
- Expertise

[Diagram: Pyramid showing the hierarchy of designing the user interface from User Model to User Documentation]

copyright 1989 Deborah J. Meyers & Associates
Managing the Design of the User Interface

**HOW DO YOU DO IT?**

**Methods**

- Scoping
- Functional Specification
- Design
- Development
- Testing/Implementation

---

**Managing the Design of the User Interface**

**HOW DO YOU DO IT?**

**Methods**

**ADDING HUMAN FACTORS TO THE SOFTWARE DEVELOPMENT PROCESS**

<table>
<thead>
<tr>
<th>Scoping</th>
<th>Functional Specification</th>
<th>Design</th>
<th>Development</th>
<th>Testing/Implementation</th>
</tr>
</thead>
<tbody>
<tr>
<td>APPLICATION</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PROJECT TEAM</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>USER INTERFACE GROUP</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

---

copyright 1989 Deborah J. Mayhew & Associates

17

10

56
Managing the Design of the User Interface

Managing the Design of the User Interface

HOW DO YOU DO IT?
Methods

ADDING HUMAN FACTORS TO THE SOFTWARE DEVELOPMENT PROCESS

<table>
<thead>
<tr>
<th>Scoping</th>
</tr>
</thead>
<tbody>
<tr>
<td>Application</td>
</tr>
<tr>
<td>Team</td>
</tr>
<tr>
<td>Business</td>
</tr>
<tr>
<td>Definition</td>
</tr>
<tr>
<td>Business</td>
</tr>
<tr>
<td>Requirement</td>
</tr>
<tr>
<td>Analysis</td>
</tr>
</tbody>
</table>

User Interface Group

Project Plan

User Profile

HW/ SW
Def's

Managing the Design of the User Interface

HOW DO YOU DO IT?
Methods

ADDING HUMAN FACTORS TO THE SOFTWARE DEVELOPMENT PROCESS

<table>
<thead>
<tr>
<th>Functional Specification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Application Project Team</td>
</tr>
<tr>
<td>Parent Spec</td>
</tr>
<tr>
<td>Task Analysis</td>
</tr>
<tr>
<td>UI Goal Setting</td>
</tr>
</tbody>
</table>

copyright 1989 Deborah J. Stephans & Associates
Managing the Design of the User Interface

**Managing the Design of the User Interface**

**HOW DO YOU DO IT?**

**Methods**

**ADDING HUMAN FACTORS TO THE SOFTWARE DEVELOPMENT PROCESS**

**APPLICATION PROJECT TEAM**
- System Architecture
- Specification
- Application Code
- UI Code Space

**USER INTERFACE GROUP**
- Corporate User Interface Standards
- Style Guide
- Detailed UI Design
- UI Prototypes
- Training and Testing

Copyright 1989 Deborah J. Mayhew & Associates
Managing the Design of the User Interface

**HOW DO YOU DO IT?**

**Methods**

**ADDING HUMAN FACTORS TO THE SOFTWARE DEVELOPMENT PROCESS**

<table>
<thead>
<tr>
<th>APPLICATION PROJECT TEAM</th>
<th>Software Testing</th>
<th>Hardware/Installation Testing</th>
<th>Production Testing</th>
</tr>
</thead>
<tbody>
<tr>
<td>USER INTERFACE GROUP</td>
<td>UI Testing</td>
<td>UI Testing</td>
<td>UI Test</td>
</tr>
</tbody>
</table>

**DJMA**

Managing the Design of the User Interface

**HOW DO YOU DO IT?**

**Expertise**

Senior Human Factors Engineer

Minimum Requirements:
- Ph.D. or Masters degree in Cognitive Psychology or Human Factors
- Minimum 2-5 years experience as a human factors professional
- Minimum 2-5 years experience in a software development environment
- Experience managing human factors/user interface projects
- Experience conducting user profiles
- Experience performing task analysis
- Experience designing user interfaces
- Experience developing style guides
- Experience designing and implementing usability tests
- Strong interpersonal skills
- Strong negotiating skills

On-The-Job Training:
- Send to short courses on advanced topics in user interface design
- Send to annual SIGCHI and Human Factors Society conferences
Managing the Design of the User Interface

**HOW DO YOU DO IT?**

**Expertise**

**User Interface Designer**

*Minimum Requirements:*
- Minimum 3-5 years experience in a software development environment
- Experience designing user interfaces
- Motivation/Interest in designing user interfaces
- Strong interpersonal skills
- Strong negotiating skills

*On-The-Job Training:*
- Send to short courses or night courses on basic user interface design and cognitive psychology
- Send to annual SIGCHI and Human Factors Society conferences

**WHY SHOULD YOU CARE?**

- Low Productivity
- High Training Costs
- Costly User Errors
- High Support Costs
- High Employee Turnover
- Underutilized Systems
Managing the Design of the User Interface

**WHY SHOULD YOU CARE?**

**Productivity**

20 Users  
230 Days per year  
80 Screens per day  
10 Seconds per screen

1022 Hrs (25.5 Wks) per year

**Training**

20 Users  
2 Systems per year  
1.5 Days per system

60 Days (12 Wks) per year
Managing the Design of the User Interface

**WHY SHOULD YOU CARE?**

**Errors**

- 600 Users
- 12 Errors per year
- 17 Minutes per error

**2040 Hrs (51 Wks) per year**

**User Support**

- 600 Users
- 4 Calls per year
- 15 Minutes per call

**600 Hrs (15 Wks) per year**
WHY SHOULD YOU CARE?

SAMPLE COST/BENEFIT ANALYSIS
OF ADDING HUMAN FACTORS TASKS
TO A SOFTWARE DEVELOPMENT PROJECT:


GENERAL ASSUMPTIONS:
Medium sized software systems (50,000 lines of source code)
15 person years to build (including HF time)
System to be used by 250 employees
Developers and HF Engineers fully loaded wages = $35.00/hour
Users fully loaded wages = $25.00/hour
Clerical support staff fully loaded wages = $15.00/hour

ANALYSIS SUMMARY:
TOTAL BENEFITS: $175,104 (per year)
TOTAL COSTS: 132,185 (one time)
FIRST YEAR SAVINGS: -$43,019

LIFECYCLE STAGE | BREAKDOWN OF COSTS
| TASK | COST | NO. | TOTAL | WEIGHT
--- | --- | --- | --- | ---
| Task Analysis | User Interviews | 4,456 | 4 | 9,000 | 1
| | User Questionnaire | 9,000 | 1 | 9,000 | 1
| | Usage Study | 2,000 | 1 | 2,000 | 1
| Design | Style Guide | 10,000 | 1 | 10,000 | 1
| | Simulation Test | 6,250 | 2 | 12,500 | 2
| | Purchase of UIMS | 16,000 | 1 | 16,000 | 1
| | Prototype Construction | 4,000 | 1 | 4,000 | 1
| | Prototype Testing | 4,250 | 2 | 8,500 | 2
| | Prototype Changes | 200 | 20 | 4,000 | 1
| System UI Testing | Prototype Test | 6,000 | 1 | 6,000 | 1
| QA | Prototype Changes | 200 | 20 | 4,000 | 1
| TOTAL COST: | 91,203 | | | 2

Managing the Design of the User Interface
Managing the Design of the User Interface

WHY SHOULD YOU CARE?

BREAKDOWN OF BENEFITS

<table>
<thead>
<tr>
<th>TYPE OF SAVING</th>
<th>AMOUNT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Decreased Training</td>
<td>$82,500</td>
</tr>
<tr>
<td>Decreased Errors</td>
<td>71,846</td>
</tr>
<tr>
<td>Increased Productivity</td>
<td>23,056</td>
</tr>
<tr>
<td>Decreased Late Design Changes</td>
<td>16,800</td>
</tr>
<tr>
<td>TOTAL BENEFITS:</td>
<td>$175,104</td>
</tr>
</tbody>
</table>

DERIVATION OF COSTS

1. HF LAB SET UP

   - Lab design and equipment selection: 160 hrs @ $35/hr  $5,600
   - Carpenters and electricians: 80 hrs @ $25/hr  2,000
   - Videocameras, VCRs, one-way mirror  12,400
   - TOTAL:  $30,000

2. USER INTERVIEWS

   - 10 Interviewees for 1 hour @ $25/hr  250
   - Interviewer @ $35/hr:
     - 16 hrs designing interview  1,800
     - 10 hrs conducting interviews
     - 28 hrs analyzing results  1,800
   - 3 Support staff @ 5 hrs each @ $15/hr  225
   - Videocassettes  60
   - TOTAL:  $3,425

64
**Managing the Design of the User Interface**

**WHY SHOULD YOU CARE?**

**DERIVATION OF COSTS**

<table>
<thead>
<tr>
<th>3. USER SURVEYS/QQUESTIONNAIRES</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Development of survey: 40 hrs @ $35/hr</td>
<td>$1,400</td>
</tr>
<tr>
<td>Pilot testing: 40 hrs @ $35/hr</td>
<td>1,400</td>
</tr>
<tr>
<td>Distribution and collection: 20 hrs @ $15/hr</td>
<td>300</td>
</tr>
<tr>
<td>Responding: 80 users for 1/2 hr @ $25/hr</td>
<td>1,000</td>
</tr>
<tr>
<td>Coding and entering data: 20 hrs @ $15/hr</td>
<td>300</td>
</tr>
<tr>
<td>Analyzing results: 40 hrs @ $35/hr</td>
<td>1,400</td>
</tr>
<tr>
<td>Computer time</td>
<td>100</td>
</tr>
<tr>
<td>Supplies and duplicating costs</td>
<td>100</td>
</tr>
<tr>
<td><strong>TOTAL:</strong></td>
<td><strong>$8,000</strong></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>4. USAGE STUDY, SIMULATION TEST OR PROTOTYPE TEST</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Development of test: 40 hrs @ $35/hr</td>
<td>1,400</td>
</tr>
<tr>
<td>Pilot testing and revisions: 40 hrs @ $35/hr</td>
<td>1,400</td>
</tr>
<tr>
<td>Running test: 40 hrs @ $35/hr</td>
<td>1,400</td>
</tr>
<tr>
<td>Subjects: 10 @ 2 hrs @ $25/hr</td>
<td>500</td>
</tr>
<tr>
<td>Analyzing results: 40 hrs @ $35/hr</td>
<td>1,400</td>
</tr>
<tr>
<td>Videotapes</td>
<td>120</td>
</tr>
<tr>
<td><strong>TOTAL:</strong></td>
<td><strong>$8,220</strong></td>
</tr>
</tbody>
</table>
### Managing the Design of the User Interface

#### WHY SHOULD YOU CARE?

**DERIVATION OF COSTS**

5. **STYLE GUIDE**

<table>
<thead>
<tr>
<th>Description</th>
<th>Cost (in $)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Author: 200 hrs @ $35/hr</td>
<td>7,000</td>
</tr>
<tr>
<td>Committee: 4 @ 50 hrs @ $25/hr</td>
<td>500</td>
</tr>
<tr>
<td><strong>TOTAL</strong></td>
<td><strong>16,500</strong></td>
</tr>
</tbody>
</table>

6. **PURCHASE OF UIMS**

<table>
<thead>
<tr>
<th>Description</th>
<th>Cost (in $)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reviewing package: 160 hrs @ $35/hr</td>
<td>5,600</td>
</tr>
<tr>
<td>Cost of average package</td>
<td>10,000</td>
</tr>
<tr>
<td><strong>TOTAL</strong></td>
<td><strong>15,600</strong></td>
</tr>
</tbody>
</table>

---

### Managing the Design of the User Interface

#### WHY SHOULD YOU CARE?

**DERIVATION OF COSTS**

7. **PROTOTYPE CONSTRUCTION** (does not include design)

<table>
<thead>
<tr>
<th>Description</th>
<th>Cost (in $)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Screen layout: 80 hrs @ $35/hr</td>
<td>2,800</td>
</tr>
<tr>
<td>Screen transitions: 80 hrs @ $35/hr</td>
<td>2,800</td>
</tr>
<tr>
<td><strong>TOTAL</strong></td>
<td><strong>5,600</strong></td>
</tr>
</tbody>
</table>

8. **PROTOTYPE CHANGE IN RESPONSE TO TESTING**

<table>
<thead>
<tr>
<th>Description</th>
<th>Cost (in $)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Screen layout: 4 hrs @ $35/hr</td>
<td>140</td>
</tr>
<tr>
<td>Screen transitions: 4 hrs @ $35/hr</td>
<td>140</td>
</tr>
<tr>
<td><strong>TOTAL</strong></td>
<td><strong>280</strong></td>
</tr>
</tbody>
</table>

---

*copyright 1998 Deborah J. Myghe & Associates*
Managing the Design of the User Interface

**WHY SHOULD YOU CARE?**

**DERIVATION OF BENEFITS**

1. **DECREASED TRAINING**
   - Typical 1 week training course reduced by 25% or 10 hrs
   - 250 users
   - Hourly rate of $25
   - 250 users X 10 hrs X $25 = $62,500 in 1st year
   - **TOTAL:** $62,500

2. **DECREASED ERRORS**
   - 250 users
   - 1.5 errors eliminated per user per day
   - 230 working days per year
   - 2 minutes in recovery time per error
   - Hourly rate of $25
   - 250 users X 1.5 errors X 230 days X $0.50 per error = $71,846 per year
   - **TOTAL:** $71,846

3. **INCREASED PRODUCTIVITY**
   - 250 users
   - 80 screens per day
   - 230 days per year
   - Processing time per screen reduced by 1 second
   - Hourly rate of $25
   - 250 users X 80 screens X 230 days X 1/600 = $23,950 per year
   - **TOTAL:** $23,950

4. **DECREASED LATE DESIGN CHANGES**
   - Changes made early cost 1/4 of changes made after implementation
   - 20 changes made early
   - 8 hrs per change
   - Hourly rate of $35
   - Early change cost = 20 changes X 8 hrs X $35 = $5,200
   - Late change cost = 4 X early change cost = $22,000
   - Savings = late change cost - early change cost = $16,800 in 1st year
   - **TOTAL:** $16,800

**Copyright 1989 Deborah J. Mayhew & Associates**
Managing the Design of the User Interface

**WHY SHOULD YOU CARE?**

- **Common questions about CBA of Human Factors function**
  - Is there a legitimate benefit for a CBA?
    - Yes
  - Time is money.
  - User time can be expensive.
  - Companies often want to increase volume of sales/service without increasing personnel.

- Will users actually take advantage of potential productivity gains?
  - Yes.
  - Studies show users realize bigger productivity gains on complex tasks than on simple ones.
  - Organizations provide incentives for productivity.
  - Personal and anecdotal experience suggest people make use of more powerful tools by being more productive.

- Why should development incur the cost when users get the benefits?
  - To meet overall business goals.
  - Development should be driven by business goals.

**WHO ELSE IS DOING IT?**

<table>
<thead>
<tr>
<th>SOFTWARE VENDORS</th>
<th>COMPUTER VENDORS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lotus</td>
<td>IBM</td>
</tr>
<tr>
<td>Apple Computer</td>
<td>DEC</td>
</tr>
<tr>
<td>Ashton-Tate</td>
<td>Wang</td>
</tr>
<tr>
<td>Microsoft</td>
<td>Xerox</td>
</tr>
<tr>
<td></td>
<td>Unisys</td>
</tr>
<tr>
<td></td>
<td>Data General</td>
</tr>
<tr>
<td></td>
<td>Hewlett Packard</td>
</tr>
<tr>
<td></td>
<td>Bell Labs</td>
</tr>
<tr>
<td></td>
<td>Symbolics</td>
</tr>
<tr>
<td></td>
<td>Sun Microsystems</td>
</tr>
<tr>
<td></td>
<td>BBN</td>
</tr>
<tr>
<td></td>
<td>AIR</td>
</tr>
<tr>
<td></td>
<td>DRC</td>
</tr>
<tr>
<td></td>
<td>GTE Labs</td>
</tr>
<tr>
<td></td>
<td>GTE Data Services</td>
</tr>
<tr>
<td></td>
<td>Mitre Corp.</td>
</tr>
<tr>
<td></td>
<td>Boeing</td>
</tr>
</tbody>
</table>

**OTHERS**

| US West          | 4 |
| Eastman Kodak    | 10 |
| Nynex            | 2 |
| CLDS Church      | 8 |

**CONTRACTORS**

| BBN              | 6 |
| AIR              | 6 |
| DRC              | 6 |
| GTE Labs         | 5 |
| GTE Data Services| 5 |
| Mitre Corp.      | 4 |
| Boeing           | 5 |
Managing the Design of the User Interface

WHAT HAS THEIR EXPERIENCE BEEN?

- An Aerospace Contractor: CAD
- A major Computer Vendor: On-Line Help
- IBM: Security
- DEC: Disk Drive
- A Phone Company: Videotex

A large aerospace contractor evaluated several CAD systems. An identical task was performed by expert operators on each vendor's equipment. Task was to input a complex 3D part, starting with engineering sketches.

RESULTS:
Fastest system: 4 hours to complete task
Slowest system: 8 hours
The contractor purchased the fastest system.
Managing the Design of the User Interface

WHAT HAS THEIR EXPERIENCE BEEN?

(A major computer vendor): On-Line Help

Engineers proposed an on-line help system be added to an existing product.
Customer Support objected, concerned it would be more software to support.
Help system was implemented.

Help calls on product reduced by 30% as a result.

GOAL:

95% end users to sign on error free after three tries.
Ideal average time for experienced user = 6 seconds.

SUBJECTS:

IBM Administrative staff
2 years experience in current jobs
Experienced computer users.

23,000 end users, large data entry and inquiry application.
Changes made to security dialog at sign on.
Managing the Design of the User Interface

WHAT HAS THEIR EXPERIENCE BEEN?

IBM: Security

RESULTS:
In Test 2, 100% signed on error free after third try
Median time on third trial in Test 2 = 7 seconds

COST/BENEFIT ANALYSIS:
Calculated for first three sign on attempts only
Total difference between Test 1 and Test 2 summed across three trials per user = 4.67 minutes
23,876 users X 4.67 minutes = 1,781 hours
Benefit = 1,781 hours X hourly rate = $41,700
Cost of three tests = $ 20,700
Net Benefit of $ 21,000
Managing the Design of the User Interface

HOW CAN YOU GET STARTED?

Short Term Action Items

Provide Education for Project Managers on Methods
Provide Education for Designers/Developers on Design Principles and Guidelines
Conduct Experimental Projects to Demonstrate Value (e.g. usability tests)
Gain Commitment

Long Term Planning
Managing the Design of the User Interface

**HOW CAN YOU GET STARTED?**

**Long Term Planning**

- Design a Human Factors methodology
- Design a Human Factors organization
- Recruit and hire Human Factors Engineers, train internal User Interface Designers
- Develop Appropriate Style Guides
- Develop/Expand a Usability Lab
- Expand the Human Factors function across the company
Technology Transfer

Marti Szczur
Goddard Space Flight Center
TAE

Yesterday, Today & Tomorrow

Marti Szczur
NASA/Goddard Space Flight Center
Software and Automation Systems Branch

Under Discussion

- TAE Classic, the Prewindow Period 1980-1985
- TAE Plus, the New Beginning, 1985-1988
- TAE Plus Matures 1989-1992
- TAE Plus Commercialization 1993
- GSFC's Future Directions
Incremental Life Cycle

Plan, Define Requirements

Design, Evaluate Alternatives

Commercial Version TAE Plus 5.3

1993

1988

1989

1983

1984

1980

1981

1982

1985

1990

1987

1991

1992

Measure, Learn, Plan Next Phases

Develop & Implement

1980

REQUIREMENTS

• multiuser, portable applications control executive

• friendly environment for users

• extensive information management support

• standard set of executive services

• extensibility to allow installation of new programs with ease

• VICAR compatible
1981

- Develop Proof-of-Concept Prototype
- Implementation language Selection Issue
- Support 3 operational systems
- Future versions must be upward compatible
- TAE Support Office created
- 1st TAE External Review

1982 - 84

- Four prototype releases between August '81 and Oct '82
- C selected for implementation language
- By 1982, 13 different projects were using prototype versions
- November 1983, first operational system, V1.0
- New releases delivered in 1983 and 1984
- Operational on VAX/VMS, PDP 11/RSX-11M, Data General Eclipse
- TAE ports into UNIX environment
- TAE Support Office works with the user community
- By 1984, 30 different user sites are recorded
1985

- Introduction to the Macintosh and the arrival of GUIs with mice and windows
- Arrival of first "low-cost" graphic workstations with windows
- 5th TAE Users' Conference
- ... And the fun begins
  - experiments with VT220 and VAXStation 100
  - the TAE Facelift phase

1986-89

- Conceptual Description of a WorkBench in '86
- Rapid Proof-of-Concept Prototype of TAE Plus in '87
  -- Smalltalk and X Windows 10
- 6th Users' Conference ('86) and 7th Users' Conference ('88)
- Object-oriented language selected for implementation language
  -- C++ or Objective C?
  -- Compiler Woes
- Papers given at ACM Symposium '87, OOPSLA '88, NCGA '88, Xhibition '89
- Two Prototypes in '88 followed by two beta releases in '89
• TAE Plus papers given at USENIX '90, MIT X Conference '90

• Over 350 Beta Test Sites

• TAE Plus V4.1 (1st Operational Release) goes to COSMIC in '90

• 8th Users Conference hosted by JSC

---

**TAE 8th Users Conference**

**Future Directions**

• Full Motif functionality support?
  -- WorkBench support for all Motif objects
  -- WorkBench support for Motif conventions/style
  -- UIL support

• *Architect/Builder* WorkBenchs

• Integrate/add object builder into TAE Plus

• Graph builder support

• Hypermedia support

• WorkBench improvements

• Support object direct manipulation and object dependencies

6/90
1990

- TAE Plus papers given at USENIX '90, MIT X Conference '90

- Over 350 Beta Test Sites

- TAE Plus V4.1 (1st Operational Release) goes to COSMIC in '90

- 8th Users Conference hosted by JSC

1991

- Honorable Mentions for "Best in Open Systems Solutions" (FEDUNIX)

- NASA Group Achievement Award to TAE Plus team

- TAE Plus presented/demoned at several aerospace conferences and tutorial at MIT X Conference

- TAE Plus Submitted as a candidate API to IEEE 1201 Committee

- V5.1 (with OSF/Motif™ toolkit) is delivered to COSMIC

- 9th TAE Users Conference in held in November '91
1992

- TAE Plus article published in The X Resource Journal
- TAE Plus presented/demoed at CHI'92 and HCI '92
- Over 500 TAE Plus V5.1 User Sites
- TAE Plus V5.2 is delivered to COSMIC
- Decision to transfer the technology...Why Now?
- Planning the transition

1993

- GSFC's TAE Project Management changes hands
- TAE Plus article published in ACM's TOIS
- V5.2 goes to COSMIC
- Technology Transfer Agreement is finalized
- TAE 10th Users Conference
- Commercialization of TAE Plus
Future Directions

Successful User Interfaces

- Usability Engineering Laboratory
- User Interface Development Tools
- HCI Guidelines Documents
- Controlled Experimental Research
  Multi-media interactions, 3-d objects

* diagram derived from Ben Shneiderman's "Three Pillars of Successful UI Design"

In Conclusion
TAE V5.3 Summary

Don Link
Century Computing Inc.
THE COMMERCIALIZATION OF TAE PLUS

Don Linz
Century Computing, Inc.
1014 West Street, Laurel, MD 20707
(301) 953-2000
Internet: dlinz@tcs.com

Agenda

- Introducing Century Computing
- TAE Plus Business Philosophy & Plans
- TAE Plus Development Directions
  BREAK
- TAE Plus v5.3
Introducing Century Computing

- Our People
- Our Company
- Our Business

Our People

- 47 Employees
- Average of 12 yrs Experience
- Expertise in C, C++, Ada
- Over 50% with Advanced Degrees
Our Company

- Founded in 1979
- Began TAE Development in 1981
- Employee Owned
- Financially Sound
- Committed to Customer Satisfaction

Our Business

Graphical User Interfaces  Image Processing
Our Business

Simulation

Spacecraft

Ground Systems

1992 Business Mix

Government Prime 18%

Commercial

Government
Business Philosophy and Plans

- Focus on Installed Base
- Emphasize Quality
- Promote Customer Participation
- Meet Customer's Needs
- Provide an Affordable Product

Focus on Installed Base

- Continue Excellent Support and Services
  - Technical Support Office
  - Newsletter
  - Users' Conference
- Make Upgrade to v5.3 Affordable
Emphasize Quality

- Fix Known Bugs
- Beta Test Sites
- New QC and QA Procedures
- Automated Testing
- Quality Service

Customer Participation

- TAE Plus Advisory Group
- Electronic Suggestion Box
- E-mail Discussion Group
- Users' Conferences
- User Surveys
- Focus Groups
Meet Customer’s Needs

- Customer Satisfaction Key to Success
- Market Forces at Work
- New Features
- Better Documentation
- Competent & Available Technical Support
- Technical Training Seminars
- Consulting Services

An Affordable Product

- Aggressive Pricing
- Creative Licensing
- Unbundle Ada and C++
- Government and Educational Discounts
Development Directions

- Truly Graphical Interfaces
- Direct Manipulation
- Rapid Prototyping and Iterative Refinement
- Non-Programmer Use
- Standards Compliance
- Integration with Other Tools
- Targeted Application Areas

Questions and Answers

Coming up:
TAE Plus v5.3
TAE PLUS v5.3

Don Link
Century Computing, Inc.
1014 West Street, Laurel, MD 20707
(301) 953-3330
Internet: dl@link.com

Development Directions

- Truly Graphical Interfaces
- Direct Manipulation
- Rapid Prototyping
- Non-Programmer Use
- Standards Compliance
- Integration with Other Tools
- Targeted Application Areas
Graphics + Direct Manipulation

- New Types of DDOs (e.g., 2-D mover)
- Import of Graphics (e.g., TIFF)
- DDO Input
- Composite DDOs
- Object Creation at Runtime
- Color Pixmaps
- Movable Items at Runtime
- Simple Graphic Decorations

2-D Movers

- Altitude (feet)
- Range (miles)
Import of TIFF Pictures

- Supports direct manipulation applications
- Increases flexibility of user input & control
• Keys are separate dynamics
• Mouse clicks on keys generate events for DDO

• Allows collections of DDOs to be treated as one unit
• Allows different types of DDOs on a common background
Runtime Creation of Dynamics

- Automated Code Merge
- GUI Scripting
- Addition of New Widgets
- Tool Maker's Workbench
- Workbench Productivity Aids

Rapid Prototyping

v5.3
Automated Code Merge

- Speeds up iterative development
- Reduces maintenance costs
- Reduces errors related to code regeneration
- Promotes iterative development

How It Works

GUI #1
- Generate Code
- Manually Edit Code
- Apply Code #1

GUI #2
- Generate Code
- Automatically Merge Code
- Apply Code #2
GUI Scripting

- Automated and Repeatable Application Tests
- Application Demos
- Tutorials
- Record/Playback Facility
- Based on perl Language

Non-Programmer Use

- Extended Connections
- Inter-Item Relationships
- Object Templates with Inheritance
- Composite Presentation Types
Standards Compliance

- v5.3
- UIL Support
- ANSI C
- Full Motif Widget Set
- Geometry Management

UIL Support

- Code Generation of UIL and Mrm
- TAE Plus Applications without Runtime Libraries (Wpt, Vm, Co)
- New DDO Widget
- Increased Portability of Applications
- Easier Access to Widget Resources & Callbacks
- Easier Use of Non-TAE Widgets
ANSI C Support

- Increased Application Portability
- Improved Code Quality and Maintenance via Function Prototypes

Integration with Other Tools

v5.3
- Software Development Tools
e.g., Energize, ObjectCenter
- U/I Design Tools
e.g., style checkers/advisors
- Usability Testing Tools
Targeted Application Areas

- Image Processing
- Geographic Information Systems
- Command & Control

Summary of TAE Plus v5.3

- 2-D Movers
- Interviews 3.1 with Graphics Import
- DDO Input
- Composite DDOs
- Runtime Creation of DDO Dynamics
- Automated Code Merge
- GUI Scripting
- UIL and DDO Widget
- ANSI C
Usability and Application Testing

Jianping Jiang
CTA Inc.

Jim Hicinbothom
CHI Systems Inc.

Sue Adams
Battelle

Phil Miller
Century Computing Inc.
PRESENTED AT

TENTH TAE USERS' CONFERENCE

JUNE, 1993

PRESENTED BY

JIANG (JIM) JIANG
GROUP ENGINEER

CTA INCORPORATED

SPONSORED BY
ADVANCED SYSTEMS PROGRAM NASA HEADQUARTERS (CODE 0)
THROUGH THE AUTOMATION TECHNOLOGY SECTION (CODE 522.3)
NASA-GODDARD SPACE FLIGHT CENTER
GREENBELT, MD 20771
WHAT IS CHIMES?

- User-Interface Designer's Associate

- Knowledge-Based Evaluation of UI Design's "Look and Feel"

- Modifier of UI Designs for Compliance with Human Factors Guidelines and Toolkit Style Guides
• PROBLEM:
How can we automate a human factors evaluation of user-interface design?

• TECHNICAL APPROACH:
- Model Demands on Users
- Check for Compliance with Guidelines
- Prototype Evaluation Concepts

• PRODUCT:
Computer-Human Interaction Models (CHIMES) Methodology and Toolset

• BENEFITS:
Savings in Time and Expense Training in Human Factors for UI Designers
• Proof-of-Concept Prototype (CHIMES '93)
  - Demonstrates evaluation graphic features for single and multiple UI panels
  - Demonstrates Advice-in-Context, including recommended colors
  - Supports automatic modification and re-evaluation of UI design
  - Supports utilization of sample-design library

• Continuing R&D In Progress
CHIMES: HIGHLIGHTS OF CURRENT AND PLANNED R&D

Current:

• Extension Of Chimes Knowledge Base

• Heuristics For Evaluation Of GUI Behavior

Planned:

• Preparation For Submission To COSMIC\(^1\) For Distribution

• Implementation Of GUI Behavior Evaluation Heuristics

• Heuristics For Evaluation Of GUI Behavior

• Integration Of CHIMES With Other UIMSs

\(^1\) COSMIC is a NASA-sponsored center for distribution of NASA software and is managed by the University of Georgia

Copies Of This Document May Be Obtained From

Walt Truszkowski  
Code 522.3  
NASA/Goddard Space Flight Center  
Greenbelt, MD  20771

301-286-8821
User Interaction Testing of the User Interface

by

James H. Hicinbothom
Senior Scientist
CHI Systems, Inc.

Sponsored by:
U. S. Army Research Laboratory
Human Research & Engineering Directorate
(Contract # DAAA15-92-C-0026)
Motivation

- Bad systems development starting with lines of code and no requirements or clear design
- Bad systems development that assumes the user is a peripheral with an I/O interface
- Slightly better systems development that incorporates evaluation, although put off until near the end of development
- Decreasing productivity after automation in some situations (e.g., when work flow obstructed by inappropriate design)
- Friends, relatives, and neighbors who are abused by bad tools at work, home, and play
Background

- Human-computer interaction dependent on situation
  \[ \text{task domain(s)} \times \text{tool(s)} \times \text{user population(s)} \]
  (task domain knowledge and tool knowledge are critical attributes of human subjects)

- Each situation different enough to constrain generalizability of interaction designs

- The only 'constant' is the human cognitive architecture, and thus, human cognitive limits

Therefore, evaluation must address both the specifics of the situation, and the more general human cognitive limits at work in the situation.

Additionally, this evaluation should be based on analyses of both the static features (e.g., layout, object semantics, vocabulary usage, and pre-defined 'connections') and the dynamic features (e.g., mappings of cognitive task structure to 'threads' of user action sequences, identified attention shifts, and undesirable navigation behaviour) of the human-computer dialogue.
Human Cognitive Limits

1. Predicting Processes
2. Combining Choice Attributes
3. Managing Information
4. Performing Intermediate Analysis/Reasoning Steps
5. Visualizing/Representing Problem and Solution Spaces
6. Making Required Judgments (e.g., Quantitative Judgment Biases, Maintaining Vigilance)
Cognitive Limits Extension Methodology

1. Perceived Need for New or Revised Interactive System
2. Situation Analysis Protocol
3. Cognitive Limits and Other Aiding/Training Constraints
4. Potential Aiding/Training Functions
5. Functional Taxonomies of Aiding/Training Techniques
6. Trade-off Rules and Criteria for Technique Selection
7. System Design Methodologies

1. Describe Problem
   - Problem Description
   - Data Description
2. Analyze Situation(s)
   - Data Review
   - Situation, Cognitive Task, and Knowledge Blackboard Models
3. Identify Limits
   - List of Applicable Limits Targeted for Aiding/Training
4. Define Requirements
   - List of Required Aiding/Training Functions
5. Identify Techniques
   - List of Candidate Aiding/Training Techniques
6. Assess Costs/Benefits
   - Situation-Defined Aiding/Training Sub-systems
7. Define Architecture

Diagram flow from left to right, with arrows indicating the sequence of steps in the methodology.
Overview of Mid-Section of Software Life-Cycle

Design
- Information requirements
- Functional and performance requirements
- Human cognitive limitations and constraints
- Implementation feedback
- Procedural design
- Architectural design
- Data design

Development
- User feedback
- User feedback
- Executable modules

Testing and Evaluation
- User-tested software
Integrating Evaluation into the Full Life-Cycle

1. Start with CLEM (Cognitive Limits Extension Methodology) for initial requirements analysis and aiding techniques selections

2. Follow up with initial architecture and design concepts, realized as Rapid Interface Prototypes using a Graphical User Interface (GUI) Builder

3. Evaluate overall architecture concept(s) and individual design concepts

4. Select or revise architecture concept (re-evaluating revisions as needed)

5. Revise initial individual design concepts (e.g., database navigator component, file selection component, etc.)

6. Define additional individual design concepts required for the chosen architecture

7. Evaluate new and revised design concepts

8. Iteratively revisit steps 5, 6, and 7, as needed

9. Integrate all available components of architecture

10. Evaluate integrated tool

11. Revise and re-integrate

12. Iteratively revisit steps 5 through 11, as needed
Tools to Aid Integrated Evaluation:
The Intelligent Interface Construction (IIICON) Evaluator

- Supports evaluation of advanced interactive systems using X Window System
- Aids Human Evaluator in preparing and managing evaluation sessions (e.g., test plan)
- Records sessions, producing both machine- and human-readable dialogue transcripts
- Records User's verbalizations, and annotations by Users and Human Evaluators
- Replays recorded sessions, including annotations, for further analyses
- Aids Human Evaluator in analyzing event sequences in dialogue
- Aids Human Evaluator in mapping semantics of dialogue
- Aids Human Evaluator in analyzing layout and organization of Graphical User Interface
- Provides a central repository for storing data, notes, and results of analyses for evaluation
- Supports distribution and re-integration of evaluation tasks, data, and results across sites
- Aids Human Evaluator in composing recommendations for design concept changes
Overview

- ATCCS Guidelines --> DoD Style Guide
- Where they fit in design process
- Impact on design
- What’s included
- Examples using TAE+
- Software demonstration
AES Methodology for User Interface Rapid Prototyping

- Interactive Prototyping
- Automated User Event Capture
- ATCCS User Interface Guidelines

AES Experimentation

- AES-Enhanced Rapid Prototyping Tools

Products for the Developer

Consistent & Usable ATCCS SMI
Commercial Style Guides

OSF/MOTIF  SUN/OPENLOOK  MICROSOFT WINDOWS

IBM PRESENTATION MANAGER  APPLE MACINTOSH

[ SELECT ONE STYLE ]

DoD HCI Style Guide

"LOOK"
"FEEL"

INTERFACE GUIDANCE
FUNCTIONALITY

Application Specific Style Guide

GENERAL

SPECIFIC
Impact on Industry and Government

- Provides positive influence on HCI design
  - Greater standardization
  - Reduced training requirements
  - Easier migration of personnel across applications and systems
  - Reduced Life-Cycle costs

- Conformance should be expected for future system design

- Need to plan for migration towards conformance
Style Guide Contents

• Discussion of differences in Motif and Open Look applications
• Hardware considerations including issues relating to Computer/Electronic Accommodation Program (CAP)
• Application Interface Design Guidelines
• Objective Security Interface Requirements
• References by Paragraph & Bibliography
Style Guide Contents

- HCI Guidelines
  - Screen Design
  - Windows
  - Menu Design
  - Object Manipulation
  - Common Features
  - Text Graphics
  - Decision Aids
  - Query
  - Embedded Training
  - Emerging Technologies
Future Directions

- Revision of Style Guide
  - Industry and Government comment
  - Style issue review
  - Uniform API (IEEE P1201.1)
  - Conformance checklist
  - "Look and Feel" (IEEE P1201.2)
  - Personal Layer

- Publication of Style Guide Version 3.0
Outline

- Motivation
- Overview
taepel Language
- Scripting Example

Note: demos available

June 16, 1993
Motivation

- Repeatable tests of TAE Plus applications
- Minimal human involvement
- End-to-end tests with automatic verification
- Stress tests
- Easily maintainable test cases
- Generation of script via "record" mode

June 16, 1993
Overview

Notes:
- Script is sequential (as opposed to event-driven)
- Script language based on perl
- Application GUI interface remains alive
- Script is "interlocked" with event handlers
- Script may control multiple applications
Based on perl; see Larry Wall's "Programming perl", O'Reilly & Associates, 1992

No perl changes; only new subroutines for application scripting

Why perl as the base?
- Public domain
- Interpreted
- C-like syntax
- Many powerful features
- Becoming very popular in the UNIX community

Note: taeperl may also be used as a GUI application language
$appHandle = &Aut'Connect ("myApplication");
for ($i=1; $i <= 1000; $i++)
{
    print "processing file number $i...
";
    &Aut'UserEvent ($appHandle, "main", "fileName", "fileNumber$i");
    &Aut'UserEvent ($appHandle, "main", "ok");
    sleep(4);
}
&Aut'Close($appHandle);
Data Driven Objects

Karl Wolf
Century Computing Inc.

Terry Bleser
Century Computing Inc.

Patricia M. Jones
University of Illinois
Tenth TAE Users' Conference '93

TAE Plus v5.3

Extensions to DDOs

Karl Wolf

Century Computing, Inc.
1014 West Street
Laurel, MD 20707
(301) 953-3330

Internet: kwolf@cen.com

June 16, 1993
Extensions to DDOs

(aka TAE Plus v5.3 DDOs)

- Refresher on current DDOs
- Plans for TAE Plus v5.3
- Open Issues

June 16, 1993
Current DDOs

- Map application data values to graphical objects.

- These objects change in response to changes in these data values in one of five ways:
  
  Mover, Rotator, Stretcher, Discrete, Stripchart

- Color thresholds can be applied to ranges of data values.

- In TAE Plus V5.2 we introduced multi-valued homogenous DDOs.

- Standard naming convention for idraw objects.

- Based on InterViews v2.6

- Entirely implemented within Wpt

June 16, 1993
Plans for TAE Plus v5.3

- New acronym DDO = Dynamic Data Object
  (was Data Driven Object)

- Implemented using InterViews v3.1

- Input Capabilities

- Extensions to DDOs

- Runtime Creation of Additional Dynamics

- Heterogeneous DDOs

- Introduction of a DDO widget
Advantages of InterViews v3.1

- InterViews v3.1 uses Glyphs. Glyphs are "light weight" objects.

- Improvements to taeidraw:
  - Can import X bitmap files as stencils.
  - Can import of color TIFF files as raster images.

- On color displays stippled fill patterns are rendered as smooth intermediate colors.
Input Capabilities

Allow user to click/drag dynamic objects

When a dynamic is changed, event handler is called

- Target value is updated (just like all other presentation types)
- Value array is filled with target Real values

Allow for click/select on dynamic objects

- Target value won't change (like Push Button)
- Event handler is called
Input Capabilities (cont.)

Input Properties apply to each dynamic in a DDO

Input Properties
- Input Enabled
- Report Events
  - At End
  - Continuously
- Feedback to User
  - Never
  - Dynamic Update
- Input Delta: [ ]

Input Delta
- Similar to Update Delta; picture and target update only if moved past input delta

Extend WptEvent structure to hold the additional information to support input (such as select vs. drag.)
Extensions to DDOs

(2-D Movers)

Each Dynamic has 2 target indices in value array

- dynamic1 => value[0], value[1] => horizontal, vertical
- dynamic2 => value[2], value[3]

Start (Range Minimum) and Stop (Maximum) Positions

- Current 1-D Mover has an implicit start (from position in idraw file)
- Current 1-D Mover has a stopn picture (or defaults to edge of DDO)
- Use same for 2-D Mover

- New "rangen" picture
- New picture named "defaultrange" (and "defaultstop")

For Multiple Dynamics wanting to use the same range (or stop)

If no associated rangen (or stopn) for a dynamic, then use defaultrange (or defaultstop)

June 16, 1993
Runtime Creation of Additional Dynamics

- With addition of 2-D Movers, naturally want to do map-type DDOs
- Allow user to create new dynamic objects from a palette (e.g. place tanks, trucks, etc.)

- Additions to the Wpt_API:
  - Wpt_CreateDynamic
  - Wpt_DeleteDynamic
Heterogeneous DDOs

- Different dynamic object types share the same static background

- In this example, dynamic3 is a discrete

- picA and picB are threshold pictures

- Still use a multi-valued Real target

- Use our v5.2 standard naming conventions to keep the WorkBench side simpler

- Input Properties, Range Min & Max, Thresholds, etc. are per Dynamic (not per DDO)

Future enhancements are planned. Terry Bleser will discuss some.
Heterogeneous DDOs (cont.)

WorkBench Details Panel (tentative)
Introduction of a DDO Widget

- The previous discussion focused on the Wpt side of the DDOs. An effort is underway to "widgetize" them.

- The plan is to support only TAE Plus v5.2 DDO functionality with the addition of multiple thresholds. This implies:
  - Each dynamic may have its own set of thresholds.
  - No input support other than what is available for current DDOs.
  - No 2-D movers.

- UIL code generation will only support v5.2 style DDOs.
Open Issues

What does Input mean to a discrete?

What does it mean to have color thresholds for an dynamic with more than one degree of freedom (e.g. 2D-Mover)?

If a stretcher has shrunk to its minimum size (0 height or width), how do you select it to stretch it out again?
Data Driven Objects
Potential Enhancements

Tenth TAE Users' Conference '93

Terry Bleser
Century Computing, Inc.
1014 West Street
Laurel, MD 20707
(301) 953-3330

Internet: tbleser@cen.com

June 16, 1993
Future Directions for DDOs

- Beyond Location, Size, and Angle
- Multi-dimensional DDOs
- Tailored Input
- Creating DDOs
- Other Media
- Escaping Flatland
Multi-dimensional Dynamics

- Move and rotate - planes, tanks
- Move and stretch (scale)
- Rotate and stretch - angle and amount for direction and speed
- Move, stretch, and rotate
- Move (rotate, stretch) and discrete

• or any combination of other attributes

June 16, 1993
Tailored Input

- Fine vs gross control
- Key/event mapping
- Hot spot definition
Tailored Input (cont'd)

- Surrogates - graphical input object pops up on select

- Gestural input
- Currently selected object
- Alternative selection methods - multi-select, cycling
- Object interactions - drag and drop, collisions

June 16, 1993
Creating DDOs

- Drawing facility
  - exact positioning and sizing
  - precise control over scaling
  - fine adjustment
  - color, font, line width assignment feedback
- Import drawings from other drawing tools
- Copy dynamics from an existing ddo - change the static only
- Arbitrary names
- Hierarchy of dynamics, groups of dynamics
- Group modify - thresholds, ranges
- Semantic attributes

June 16, 1993
Other Media

Sound output
  data representation - scatter plot, size of mail message
  symbolic "picture"
  interaction of objects

Voice input

June 16, 1993
Escaping Flatland

- 2-d Stretcher
- 3-d Mover, stretcher, rotator
- 3-d Surrogates - shadows

June 16, 1993
User Experiences with Data-Driven Objects

Patricia M. Jones

University of Illinois at Urbana-Champaign

Department of Mechanical and Industrial Engineering
1206 W. Green St.
Urbana IL 61801

TAE Plus User's Conference, June 1993
• Part of "pilot" course on Interactive Systems Design

Application: Manufacturing

Students' favorite Presentation Item: Discrete DDO

Very flexible

Utilized example in documentation on switching picture files dynamically

Easy to use

Fun!
Wish List

Generate code for thresholds set for DDO's

(e.g.,

#define ITEM1_PICTURE1_THRESHOLD 10

better yet:

#define ITEM1_RED 10)

Incorporate dynamic text into Movers (e.g., for AGV)

In general, composite DDOs
New DDO: "Tracker" object for vehicle applications.

User draws arbitrary path ("static"), "dynamic" vehicle follows it.
Object Dependencies

Craig Warsaw
Century Computing Inc.

Margi Klemp
University of Colorado
Tenth TAE Users' Conference '93

Object Dependencies

Craig Warsaw
Century Computing, Inc.
1014 West Street
Laurel, MD 20707
(301) 953-3330

Internet: cwarsaw@cen.com

June 16, 1993
Overview

Goals

Static Layout vs. Dynamic Behavior

Current Connection Capability

Separate User Interface from Application

Miscellaneous Connection Enhancements

Item-to-Item Connections

June 16, 1993
Goals

Enhancements to Connections

Allow non-programmer/UI designer to specify more dynamic behavior

Improve separation of UI and application (potentially)

Interested in input from the user community

These enhancements are only in the idea phase. They are not slated for a particular release

June 16, 1993
Non-programmer uses the WorkBench to define the User Interface

Static Layout is composed of static layout and dynamic behavior

- Visual Attributes, Position, Size, Color, Font, Label
- DDO Thresholds, Connections

Dynamic Behavior

Problem: Most UI dynamic behavior must be specified by the programmer.
Programmer should only be concerned about the application, not the UI.

Solution: Extend connections to allow non-programmer to specify more dynamic behavior
Current Connection Capability

Item-to-Panel

Connections allow non-programmer to specify simple dialog control, e.g.
- When the user presses buttonA on panel1, panel2 appears and panel1 disappears
- Display a different panel for each choice in a menu

No Separation of User Interface from Application
- Connection is implemented via code generated into the event handler
- If connection is changed (in the WorkBench), code must be regenerated and recompiled

June 16, 1993
Separate User Interface and Application

To Generate into Code -- Or Handle in Wpt

Generate connections into code

- Generated code can be modified (by programmer) to integrate dynamic behavior with application knowledge
- E.g. if database is empty, display panel1, else display panel2

Handle connections in Wpt

- Change connection in WorkBench doesn’t require application change
- Should be able to change connection at run-time

Which would most fit people’s needs

- Applies to current and future connections
- An option would be most desirable, but we need to focus our development efforts

June 16, 1993
Miscellaneous Connection Enhancements

Item-to-Multiple Panels
Create, delete, or change state of many panels from a single event

Handle Multiple Selection - Selection List
Loop through all indices of value array (must be done for all event handlers)

Default Connection
A single connection for all choices of a multiple connection item
E.g. Show panel 2 when any choice is made from a radio button

MACRO facility - developed by University of Colorado
Allows non-programmers to insert code using customizable macros

Request input from user community

June 16, 1993
Item-to-Item Connections

Potential types of item-to-item connections (all driven by user-events)

- Update Properties (e.g. Sensitivity, Visibility)
  Select a mode from a Radio Button — certain control buttons dim

- Update Target values
  Manipulate a Scale — set the target value of a DDO

- Update Constraints or Menubar entries
  Press a button — change the choices of a menu

- Update View attributes
  Check a checkbox — Change the title and foreground color of a label

Request input from user community

- Consider your applications — How often would this be useful?

- Even without application knowledge?
USING THE SPREADSHEET MODEL OF COMPUTATION FOR DEFINING OBJECT DEPENDENCIES

PRESENTED BY

MARGI KLEMP
UNIVERSITY OF COLORADA
Using the Spreadsheet Model of Computation for Defining Object Dependencies

Why the Spreadsheet?

• Programming languages are difficult for non-specialists

• The spreadsheet model has done more to make computing accessible than any development since Fortran (Clayton Lewis - New Approaches to Programming, 1989)

• The spreadsheet model fits well with graphical user interfaces which can be viewed as intercommunicating objects

• NoPumpG extends the spreadsheet model to control graphical interactions and animation (Lewis, 1987)

• Software development projects at the University of Colorado are building on this model to define interactions of objects used for scientific visualization
Spreadsheet View of TAE objects:

<table>
<thead>
<tr>
<th>Object Type</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
<th>G</th>
<th>H</th>
<th>I</th>
<th>P</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bg Color</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Fg Color</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Font</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Xsize</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ysize</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Visible</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Xorigin</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Yorigin</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Geometry Management Examples

Assume the user resizes the panel:

• Item B (rotator) changes size in proportion to the new window size

  \[ B.Xsize = 0.2 \times P.Xsize \]
  \[ B.Ysize = 0.1 \times P.Ysize \]

• The position of B remains the same relative to the new panel size

  \[ B.Xorigin = 0.1 \times P.Xsize \]
  \[ B.Yorigin = 0.05 \times P.Ysize \]

• Item E (button) remains the same size regardless of panel size. There are no formulas for the size cells

  \[ E.Xsize = 50 \]
  \[ E.Ysize = 20 \]
Geometry Management Examples (continued)

- Items E - H (buttons) are always displayed in the same order but will be placed in separate rows if not fully visible on the panel

  \[ E.X\text{origin} = .1 \times P.X\text{size} \]

  F - H are positioned relative to the previous button.
  We create an ordinary cell for the previous X distance.

  \[ \text{previous-diste} = E.X\text{origin} + E.X\text{size} + \text{spacex} \]

  The formula for the X origin of button F is an if construct:

  \[ F.X\text{origin} = \text{if}(F.X\text{size} + \text{previous-diste} > P.X\text{size}, E.X\text{origin}, \text{previous-diste}) \]

- Object visibility could be controlled by a formula. Assume that item D (graph) should be invisible if the panel X size is less than 180

  \[ D.\text{visible} = \text{if}(P.X\text{size} < 180, 0, 1) \]

- If an item on a panel were resized, the origin and size of surrounding items could be defined in terms of the new size of adjacent items
Other Examples

- Attributes can be propagated via formulas. To maintain the same background color for buttons E - H define an ordinary cell for the color

  \[
  \text{Button-color} = \text{"red"}
  \]

  Then use formulas to set the color for each button

  \[
  \text{E.Bgcolor} = \text{Button-color}
  \]
  \[
  \text{F.Bgcolor} = \text{Button-color}
  \]
  \[
  \ldots
  \]

  Note that the color could be set from a menu item, a text list, etc.

- A checkbox (J) could control the visibility of workspace I

  \[
  \text{I.visible} = \text{J}
  \]
A Simpler Interface for the TAE WorkBench

Panel resize options:

- Resize an item (or all items on panel) in proportion to the new panel size
- Leave an item (or all items on panel) the same size clipping where necessary
- Group selected items maintaining sequential positioning within the group (create extra rows or columns as needed)

Specify the options above via standard TAE interaction objects

Automatically generate the spreadsheet including formulas to define the selected option
Summary

- The spreadsheet model of computation appears to handle many of the problems encountered by user interface designers in regard to object dependencies which would traditionally require a programming solution.

- Formulas may become quite complex. A simpler interface could be used to define the behavior for the most commonly used scenarios.
Integration with Other Software

Chris Barclay, Joseph Molnar
Naval Research Lab.

Ken Sall
Century Computing Inc.

Greg Shirah
Goddard Space Flight Center
The Development of a Graphical User Interface to the Fault Isolation System Database Manager

Delivered to the Tenth TAE User's Conference
June 14-17, 1993

Christopher Barclay
Joseph Molnar

Information Technology Division
Naval Research Laboratory
Enhanced knowledge database development and management

**Goal:** Working Expert System
- Reliability
- Ease of Use
- Robust
- Data Management

**Method:**
- Empress
  - Reliable Database
  - Data Management Functionality
- TAE
  - Ease of Use
  - Rapid Prototyping
  - Intuitive Interface
Choose a selection:
^ Database Conversion
^ Print a Database
^ Expert Knowledge Aquisition
^ Work with a Database
^ Create a Database
^ Delete a Database

[OK]  [Help]  [Quit]
What type of database is it:
\^ Rules
\^ Tests

Current Files:
bugs
close.c
compile*
fisdmpro*
fisdmpro.c
defompio.clog
\underline{fisdmpro.h}
fisdmpro.mak
fisdmpro.o

What is the database you would like to convert
\underline{fisdmpro.h}

What is the name of the new file

Convert  Main Menu
Move the mouse and click to enter values.

What is the module name: a26a1a1_delay_line

What is the cause: a26a1a22_a17-out beam_3_wave bad

What is the effect: a26a1a1J4 bit_0_left_input bad

What is the type: D

What is the precondition: t

What is the failure rate: 0.1

Skip:

Enter number of records to skip 2

Enter the name

OK
Tenth TAE Users' Conference '93

UIL Support and
Mrm Code Generation

Kenneth B. Sall
Century Computing, Inc.
1014 West Street
Laurel, MD 20707
(301) 953-3330

Internet: ksall@cen.com

June 16, 1993
UIL Support and Mrm Code Generation

**Overview**

- What are UIL and Mrm?
- Advantages of UIL/Mrm Applications
- Advantages of Wpt Applications
- UIL Generation
- Sample Mrm Code (prototype)
- Sample UIL File (prototype)
What are UIL and Mrm?

UIL is Motif's *User Interface Language*

- Permits separation of user interface specification from application code.
- Textual description of the UI which is compiled into binary form called **UID** (*User Interface Definition*) using the Motif compiler, named *uils*.
- *Static* description (e.g., no item-to-panel connections)

MRM is the *Motif Resource Manager*

- Set of functions in libMrm.a which retrieve the widget hierarchy from the UID file and create the widgets.
- Application code defines callbacks in the normal X Toolkit manner, but doesn't call XtCreate[Managed]Widget.
Advantages of UIL/Mrm Applications

- A more **standard representation** for interfaces developed with TAE Plus.

- **No proprietary libraries** (DDOs, however, require new DDO widget library, libXtae.a).¹

- Eliminates the requirements for applications to use the Wpt, Vm, and Co runtime packages, thereby significantly reducing the **size of executables**.²

- Eases the **porting** of applications to platforms not supporting TAE Plus.

- New interfaces developed in TAE Plus will be more easily migrated to **other UIDs** (user interface development tools).

---

¹ Link libs are simply "-[l|d|t]eae -i|nter|viewsX11] -lMr|m -lXm -lXt -lX11" instead of "-lwpt -lterm -lddo -lw|m -lInter|viewsX11 -lXm -lXt [-lXmu] -lX11 -ltae -ltae -ltermlib -lm -lc"

² Test case: single panel with 29 items [all presentation types except DDOs, color logger, and dynamic text]; static layout only; comparison of Sun stripped binary size. **UIL application was approximately one-half the size of the Wpt version.** (Size of interface description files was approximately the same.)
Advantages of UIL/Mrm App. (cont.)

- Improved application **performance** using compiled UIL file (app.uid).

- Permits access to all **widget resources and callbacks** for finer control than is allowed in the WorkBench.  

- Enables addition of **widgets not supported** by TAE Plus. Knowledgeable Motif programmers can directly add Motif widgets (e.g., XmArrowButton, XmScrollbar, XmCommandBox) to app.uil.

- All 23 Presentation Types supported including DDOs.

  **Note:** To use UIL, your Motif vendor must supply the Mrm library (default location: /usr/lib/libMrm.a) and the **uil** compiler (default: /usr/bin/X11/uil). Most vendors do provide these.

---

3. At this time, automatic merging of hand-edits to generated UIL when regenerating is still TBD.
4. Can also add your own widgets by registering them with UIL, which is what we've done with DDOs. This will be covered in the v5.3 **Guidelines for Adding a New Presentation Type**.
5. Dynamic Text is generated as simply an XmLabel widget in v5.3.

---

*June 16, 1993*
Advantages of Wpt Applications

- Wpt library provides greater functionality, usually with less application code than Xt, whereas UIL apps. have to simulate Wpt_PanelMessage, Wpt_HideItem, Wpt_ParmReject, etc.

- Designer and programmer need not be as familiar with Motif, Xt, and Xlib details, especially Motif resources and callbacks.

- **Automatic error checking**, such as for constraints (e.g., keyin, multi-line edit)

- Customized error messages (keyin, multi-line edit, textlist)

- TAE Plus Help mechanism

- **Scripting** (a v5.3 feature) - recording and playing back

- **Code merging** (a v5.3 feature; TBD whether in v5.3 UIL)

- Item-to-panel connections may only be available to Wpt applications. (TBD whether supported for UIL in v5.3.)
UIL Generation

- Generated code is Mrm, Xt, and some Xlib (no Wpt or IV).
- Type "make" to build; type "make app.uid" to compile app.uid (into app.uid) using the uil compiler.
Sample Mrm Code (prototype)

/* *** TAE Plus Mrm Code Generator version 5.3 *** */
#include <stdio.h>
#include <Mrm/MrmPublic.h>
#include <Xm/Xm.h>
#include <X11/Intrinsic.h>
#include <X11/StringDefs.h>
#include <Xm/MwmUtil.h> /* for MWM_DECOR_* and MWM_FUNC_* */
#define MAX_ARGLIST 12

int SetTopLevelResources()
{

Display *TheDisplay;
XtApplicationContext AppContext;
Widget TopLevelWidget;
MrmHierarchy S_MrmHierarchy;

int main (argc, argv)
int argc;
char **argv;
{
    int n;
    Arg arglist[MAX_ARGLIST];
    MrmType dummy_class;
    Widget main_window_widget = NULL;
    static char *db_filename_vec[] = {"app.uid");
    static int db_filename_num =
        (sizeof db_filename_vec / sizeof db_filename_vec[0]);
}
MrmlInitialize();
XtToolkitInitialize();
AppContext = XtCreateApplicationContext();
TheDisplay = XtOpenDisplay(AppContext, NULL, argv[0], "theApplication",
    NULL, 0, &argc, argv);
if (TheDisplay == NULL) {
    fprintf (stderr, "%s: Can't open display\n", argv[0]);
    exit (1);
}

n = SetTopLevelResources (arglist, "Presentation Types Demo",
    617, 871, 511, 39, 5,"presdemo", "/$TAE/inc/bitmaps/tae.icon",
    (MWM_DECOR_MENU|MWM_DECOR_TITLE|MWM_DECOR_BORDER|
    MWM_DECOR_TITLE|MWM_DECOR_MINIMIZE),
    (MWM_FUNC_MOVE | MWM_FUNC_MINIMIZE ));

TopLevelWidget = XtAppCreateShell("ToppresdemoPanel", NULL,
    applicationShellWidgetClass,
    TheDisplay, arglist, n);

if (MrmOpenHierarchy (db_filename_num, /* Number of files. */
    db_filename_vec, /* Array of file names. */
    NULL, /* Default OS extension. */
    &S_MrmHierarchy) /* Pointer to returned MRM ID */
! =MrmsSUCCESS)
{
    fprintf (stderr, "can't open hierarchy\n");
}
exit (1);
}
RegisterCallbacks ();

if (MrmFetchWidget (S_MrmHierarchy,
    "presdemoPanel", /* uil name of panel */
    TopLevelWidget, /* TBD */
    &main_window_widget, /* TBD */
    &dummy_class) /* TBD */
    != MrmSUCCESS)
{
    fprintf (stderr, "can't fetch main window\n");
    exit (1);
}

XtManageChild (main_window_widget);
XtRealizeWidget (TopLevelWidget);
XtAppMainLoop (AppContext);
} /* main */

void presdemo_textlist_cb (widget, client_data, call_data)
{
    Widget widget;
    XtPointer client_data;
    XtPointer call_data;
    {
        printf ("event handler: presdemo/textlist\n");
    }
void presdemo_checkbox_cb (widget, client_data, call_data)
    Widget widget;
    XtPointer client_data;
    XtPointer call_data;
    {
        printf ("event handler: presdemo/checkbox\n");
    }

    /* list of functions to register */

    static MrmRegisterArg RegList[] =
        {
            ("presdemo_textlist_cb", (XtPointer)presdemo_textlist_cb),
            ("presdemo_checkbox_cb", (XtPointer)presdemo_checkbox_cb),
            ("", 0) /* dummy last entry */
        };
    #define NRegList (sizeof(RegList)/sizeof(RegList[0]) - 1)

    int RegisterCallbacks ()
    {
        int code;
        code = MrmRegisterNames (RegList, NRegList);
        if (code != MrmSUCCESS)
            {
                printf ("cannot register callbacks\n");
                return;
            }
    } /* RegisterCallbacks */
int SetTopLevelResources ( arglist, title,
    width, height, x, y, border,
    iconName, iconFilename,
    decorMask, funcMask )

Arg arglist[];
char *title;
Dimension width, height;
Position x, y;
Dimension border;
char *iconName, *iconFilename;
unsigned long decorMask, funcMask;

{ int n = 0;
    XtSetArg (arglist[n], XmNtitle, title ); n++;
    XtSetArg (arglist[n], XmNmwmDecorations, decorMask ); n++;
    XtSetArg (arglist[n], XmNmwmFunctions, funcMask ); n++;
    XtSetArg (arglist[n], XmNwidth, width ); n++;
    XtSetArg (arglist[n], XmNheight, height ); n++;
    XtSetArg (arglist[n], XmNx, x ); n++;
    XtSetArg (arglist[n], XmNy, y ); n++;
    XtSetArg (arglist[n], XmNiconName, iconName ); n++;
    XtSetArg (arglist[n], XmNborderWidth, border ); n++;
    return ( n ); /* number of resources set in arglist */
} /* SetTopLevelResources */
Sample UIL File (prototype)

module main
version = 'v1.1'
names = case_sensitive

    pixmap_icon: xbitmapfile('/net/bat/home/tae/v53/inc/bitmaps/tae.icon');
color_black: color('black');
color_gold: color('gold');
font_alias_courB18: font('courB18');
procedure
    presdemo_textlist_cb();
presdemo_checkbox_cb();

object presdemo_checkbox : XmToggleButton
    {
        arguments
            {
        ! Item Specification Panel resources
            XmNlabelString = "Checkbox";
            XmNfontList = font_alias_courB18;
            XmNx = 15;
            XmNy = 26;
            XmNwidth = 129;
            XmNheight = 53;
            XmNforeground = color_black;
            XmNbackground = color_gold;
            XmNborderColor = color_black;
            XmNborderWidth = 2;
XmNshadowThickness = 2;
XmNtraversalOn = true;
XmNnavigationType = XmTAB_GROUP;
XmNhilightThickness = 0;
! Presentation Panel resources
XmNspacing = 5;
XmNset = true;
XmNalignment = XmALIGNMENT_BEGINNING;
XmNselectColor = color_black;
};
callbacks
{
    XmNvalueChangedCallback = procedure presdemo_checkbox_cb();
};

object presdemo_textlist : XmBulletinBoard
{
    arguments
    {
        ! Item Specification Panel resources
        XmNx = 446;
        XmNy = 94;
        XmNwidth = 117;
        XmNheight = 112;
        XmNforeground = color_blue;
        XmNbackground = color_beige;
        XmNborderColor = color_blue;

        June 16, 1993
XmNborderWidth = 1;
XmNshadowThickness = 2;
XmNtraversalOn = true;
XmNnavigationType = XmTAB_GROUP;
XmNhighlightThickness = 2;
! Presentation Panel resources
XmNmarginHeight = 0;
XmNmarginWidth = 0;
};
controls
{
XmScrolledList
{
arguments
{
XmNx = 1;
XmNy = 22;
XmNwidth = 117;
XmNheight = 89;
XmNfontList = font_alias_variable;
XmNforeground = color_blue;
XmNbackground = color_beige;
XmNborderColor = color_blue;
XmNitemCount = 5;
XmNitems = string_table( "choice 1",
   "choice 2",
   "choice 3",
   "choice 4",
   "choice 5" );
XmNvisibleItemCount = 3;
XmNselectedItemCount = 3;
XmNselectedItems = string_table("choice 2",
"choice 3",
"choice 4");
XmNselectionPolicy = XmMULTIPLE_SELECT;
XmNlistSizePolicy = XmVARIABLE;
XmNscrollBarDisplayPolicy = XmSTATIC;
XmNhilightThickness = 0;
XmNlistMarginHeight = 3;
XmNlistMarginWidth = 3;
XmNlistSpacing = 3;
);
callbacks
{
XmNmultipleSelectionCallback = procedure presdemo_textlist_cb();
};

XmLabel
{
arguments
{
XmNx = 2;
XmNy = 4;
XmNheight = 14;
XmNwidth = 117;
XmNfontList = font_alias_variable;
XmNforeground = color_blue;

June 16, 1993
XmNbackground = color_beige;
XmNborderColor = color_blue;
XmNlabelString = 'Selection List';
XmNalignment = XmALIGNMENT_BEGINNING;

};

};

};

object presdemoPanel : XmBulletinBoard
{
    arguments
    {
        XmNx = 511;
        XmNy = 39;
        XmNwidth = 617;
        XmNheight = 871;
        XmNforeground = color_black;
        XmNbackground = color_beige;
        XmNborderColor = color_black;
        XmNborderWidth = 5;
        XmNresizePolicy = XmRESIZE_GROW;
        XmNtraversalOn = false;
        XmNnavigationType = XmNONE;

        ! TBD: Font default for objects without font resources
        XmNlabelFontList = font_alias_variable;
        XmNbuttonFontList = font_alias_ncen14;
    }

June 16, 1993
Greg Shirah
Code 522

NASA - Goddard Space Flight Center
<table>
<thead>
<tr>
<th>Experience</th>
<th>CenterLine's Object Center C++ Compiler with TAE</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>C++</strong></td>
<td>2 years</td>
</tr>
<tr>
<td><strong>TAE+</strong></td>
<td>1 year (no TAE C experience)</td>
</tr>
<tr>
<td><strong>X/Motif</strong></td>
<td>3+ years</td>
</tr>
</tbody>
</table>
GenSAA

CenterLine's Object Center C++ Compiler with TAE

ObjectCenter C++

TAE

TPOCC (Transportable Payload Operations Control Center)

Other

GenSAA (Generic Spacecraft Analyst Assistant)
What is GenSAA?

- Generic Spacecraft Analyst Assistant
- Graphical expert system builder for spacecraft monitoring & fault isolation
- Written in Centerline’s C++
- Used TAE 5.2 for GenSAA Workbench
- Integrated with TPOCC
- GenSAA Workbench - graphical specification of:
  - Data to be monitored/generated
  - Expert System Rules
  - User Interface
- GenSAA Runtime - execution environment
Object Center C++ With TAE

- Object Center interprets source or loads object code
- Used graphical debugger initially
- Found bug in Object Center related to displaying TAE widgets - Object Center / TAE responded with a fix
- Our system grew to be too large to load into Object Center's debugger
Using TPOCC with TAE and ObjectCenter C++

- TAE & TPOCC redefine several common macros
  - LONG
  - DOUBLE
- TPOCC used a C++ keyword "class"
- TPOCC & TAE work together smoothly, otherwise
Lessons Learned

- Object Center is very good at:
  - Enabling quick access to source files
  - Identifying compile time errors
  - Identifying runtime errors
  - Unit testing

- Object Center is not so good at:
  - Debugging large systems
Terry Bleser  
Century Computing Inc.  
1014 West Street  
Laurel, MD 20707

Sharon A. Braveman  
Computer Sciences Corp.  
4600 Powder Mill Road  
Beltsville, MD 20705

P C Caldwell  
Computer Sciences Corp.  
4600 Powder Mill Road  
Beltsville, MD 20705

David Carr  
Goddard Space Flight Center  
Code 520.9  
Greenbelt, MD 20771

Leslie Carter  
CTA Inc.  
6116 Executive Blvd.  
Suite 800  
Rockville, MD 20852

Sharon Chapman  
Naval Undersea Warfare Center  
610 Dowell Street  
Code 4322  
Keyport, WA 98345

Jay J. Chen  
Allied Signal Tech Service  
Code 562.7  
Greenbelt, MD 20771

John G. Bury  
Computer Sciences Corp.  
4600 Powder Mill Road  
Beltsville, MD 20705

Melana Clark  
Goddard Space Flight Center  
Code 522  
Greenbelt, MD 20771

Kathryn Cables  
Naval Surface Warfare Center  
Dept. K51  
Dahlgren, VA 22448

Eric Davis  
Naval Surface Warfare Center  
Code 214  
Carderock Division  
Bethesda, MD 20084

Yun Jia Cai  
Science Application & Systems Inc./GSFC  
Code 912  
Greenbelt, MD 20771
<table>
<thead>
<tr>
<th>Name</th>
<th>Company</th>
<th>Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stowe Davison</td>
<td>Century Computing Inc.</td>
<td>1014 West Street, Laurel, MD 20707</td>
</tr>
<tr>
<td>M. &quot;Shelly&quot; Dumas</td>
<td>Fairchild Space</td>
<td>20301 Century Blvd, Germantown, MD 20874</td>
</tr>
<tr>
<td>Bob Deen</td>
<td>JPL</td>
<td>4800 Oak Grove Drive, Mail Stop 168-414,</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Pasadena, CA 91109</td>
</tr>
<tr>
<td>Lisa A. Dutra</td>
<td>Carlow International Inc.</td>
<td>3141 Fairview Park Drive, Suite 750,</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Falls Church, VA 22042</td>
</tr>
<tr>
<td>Valerie Denney</td>
<td>Martin Marietta Civil Space &amp; Communications</td>
<td>4451 Parlament Place Suite C, Lanham, MD 20706</td>
</tr>
<tr>
<td>Debbie Eckel</td>
<td>Naval Surface Warfare Center</td>
<td>Code K51, Dahlgren, VA 22448</td>
</tr>
<tr>
<td>Steven L. Dodge</td>
<td>Naval Surface Warfare Center</td>
<td>Code K52, Dahlgren Road, Dahlgren, VA 22448</td>
</tr>
<tr>
<td>Steve Edwards</td>
<td>Goddard Space Flight Center</td>
<td>Code 522, Greenbelt, MD 20771</td>
</tr>
<tr>
<td>Wayne Donavan</td>
<td>Vitro Corp.</td>
<td>P.O. Box 381, Dahlgren, VA 22448</td>
</tr>
<tr>
<td>Kathleen A. Erno</td>
<td>Computer Sciences Corp.</td>
<td>4600 Powder Mill Road, Beltsville, MD 20705</td>
</tr>
<tr>
<td>Edward Dorsie</td>
<td>Booz Allen &amp; Hamilton</td>
<td>7404 Executive Place, Suite 500,</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Seabrook, MD 20706</td>
</tr>
<tr>
<td>Shin-tsuen Fang</td>
<td>ATSC</td>
<td>12015 Citrus Grove Road, N. Potomac, MD 20878</td>
</tr>
<tr>
<td>Deborah Douglass</td>
<td>Goddard Space Flight Center</td>
<td>Code 683.2, Greenbelt, MD 20771</td>
</tr>
<tr>
<td>Mary Farrall</td>
<td>Computer Sciences Corp.</td>
<td>4600 Powder Mill Road, Beltsville, MD 20705</td>
</tr>
</tbody>
</table>
David Fout  
Century Computing Inc.  
1014 West Street  
Laurel, MD  20707

Bernard Harris  
Goddard Space Flight Center  
Code 532.3  
Greenbelt, MD  20771

Darren Gemoets  
Century Computing Inc.  
1014 West Street  
Laurel, MD  20707

Elfrieda Harris  
Goddard Space Flight Center/RMS  
Code 522  
Greenbelt, MD  20771

John Gillen  
Century Computing Inc.  
4141 Col. Glenn Highway  
Dayton, OH  45431

Jonathan Hartley  
Goddard Space Flight Center  
Code 520  
Greenbelt, MD  20771

Christine Goertz  
DLR/GSOC  
SEEFELDER WEG  
8031 Gilching  
GERMANY

Jim Hicinbothom  
CHI Systems Inc.  
Gwynedd Plaza III  
Bethlehem Pike at Sheble Lane  
Spring House, PA  19477

Nancy Goodman  
Goddard Space Flight Center  
Code 522  
Greenbelt, MD  20771

Jean Hobgood  
NSWC DD  
Code (K51)  
Dahlgren, VA  22448

Thomas E. Greenhorn  
Goddard Space Flight Center/ATSC  
B23 Rm. W153  
Greenbelt, MD  20771

Jackie Hoglund  
Computer Sciences Corp.  
4600 Powder Mill Road  
Beltsville, MD  20705

Kimberly A. Gudyka  
NSWC DD  
Code (K51)  
Dahlgren, VA  22448

David Howell  
Goddard Space Flight Center  
Code 520  
Greenbelt, MD  20771
George J. Huffman  
Goddard Space Flight Center/USRA  
Code 912  
Greenbelt, MD 20771

Dennis Kingsley  
Naval Sea Systems Command  
Code O5H3  
Washington, DC 20362

Pete Hughes  
Goddard Space Flight Center  
Code 522  
Greenbelt, MD 20771

Margi Klemp  
University of Colorado  
1234 Innovation Drive  
Boulder, CO 80303

David Jennings  
Naval Surface Warfare Center  
Code K51  
Dahlgren, VA 22448

Connie Kroneman  
Goddard Space Flight Center  
Code 541.2  
Greenbelt, MD 20771

Jim Jiang  
CTA Inc.  
6116 Executive Blvd. Suite 800  
Rockville, MD 20852

Chris Langford  
Coastal Systems Station  
Code 3120  
Panama City, FL 32407

Patricia Jones  
University of Illinois  
140 Mechanical Engineering Bldg.  
1206 West Green Street  
Urbana, IL 61801

Doug Lankenau  
Goddard Space Flight Center/ATF  
Code 513  
Greenbelt, MD 20771

Stephen H. Jovanelly  
Goddard Space Flight Center/RMS  
Code 522  
Greenbelt, MD 20771

Kathleen P. Le  
U.S. Army CAA  
8120 Woodmont Avenue  
Bethesda, MD 20814

Patricia W. Kilduff  
U.S. Army Research Lab.  
AMSRL-HR-SA (Kilduff)  
Aberdeen Proving Ground, MD 21005-5425

Jae Lee  
National Weather Service  
1325 East-West Highway  
Room 16303  
Silver Spring, MD 20910
Kuang-Tzung Liang
Hughes STX
4400 Forbes Blvd.
Lanham, MD 20706

Connie Liang
The Nature Conservancy (TNC)
1815 N Lynn Street
Arlington, VA 22209

Gene Liberman
Sverdrup Corp.
2001 Aerospace Parkway
Brookpark, OH 44142

Don Link
Century Computing Inc.
1014 West Street
Laurel, MD 20707

Karen Lolio
Computer Sciences Corp.
4600 Powder Mill Road
Beltsville, MD 20705

Keith Lukshin
Computer Sciences Corp.
6100 Western Place
Suite 105, Mail Stop 5303
Ft. Worth, TX 76107

Art Manksy
Vitro Corporation
1400 Georgia Avenue
Silver Spring, MD 20906

Mary Martz
Computer Sciences Corp.
4600 Powder Mill Road
Beltsville, MD 20705

Deborah Mayhew
Deborah Mayhew Associates
Panhandle Road
P.O. Box 248
West Tisbury, MA 02575

John McBeth
Century Computing Inc.
1014 West Street
Laurel, MD 20771

Claudia McIlwain
Century Computing Inc.
1014 West Street
Laurel, MD 20707

Mary Meckley
HRB Systems
P.O. Box 60
Science Park Road
State College, PA 16804-0060

Johnny E. Medina
Goddard Space Flight Center
Code 522
Greenbelt, MD 20771

Cindy Mellor
Naval Surface Warfare Center
Code K51
Dahlgren, VA 22448
Phil Miller  
Century Computing Inc.  
1014 West Street  
Laurel, MD 20707

Karen Moe  
Goddard Space Flight Center  
Code 522  
Greenbelt, MD 20771

Michael Mohler  
Loral AeroSys  
7375 Executive Place, Suite 101  
Seabrook, MD 20706

Walt Moleski  
Goddard Space Flight Center  
Code 522.2  
Greenbelt, MD 20771

Joseph Molnar  
Naval Research Lab.  
4555 Overlook Avenue  
Code 5524  
Washington, DC 20375-5000

Betty Murphy  
CTA  
6116 Executive Blvd., Suite 800  
Rockville, MD 20852

John J. Nelka  
Advanced Marine Enterprises, Inc.  
2341 Jefferson Davis Highway  
Arlington, VA 22202

Cuong M. Nguyen  
Naval Surface Warfare Center  
10901 New Hampshire Avenue  
Code B40  
Silver Spring, MD 20903-5640

Deanna Niechwiadowicz  
HRB Systems  
P.O. Box 60 Science Park Road  
State College, PA 16804-0060

Michael Nowak  
Naval Surface Warfare Center  
Code L13U  
Dahlgren, VA 22448

Christopher Olson  
Logicon  
1408 Fort Crook Road South  
Bellevue, NE 68005

Paul Paluzzi  
Computer Sciences Corp.  
4600 Powder Mill Road  
Dept. 562  
Beltsville, MD 20705

Steve Panzer  
Century Computing Inc.  
1014 West Street  
Laurel, MD 20707

Steve Pardue  
International Software Systems  
Suite 250, Echelon N  
9430 Research Blvd.  
Austin, TX 78759
Jigish Patel  
Computer Science Corp.  
4600 Powder Mill Road  
Beltsville, MD  20705

Linda Rosenberg  
Goddard Space Flight Center  
Code 522  
Greenbelt, MD  20771

Ajay D. Pattni  
Comptek Federal Systems Inc.  
2121 Crystal Drive  
Suite 700  
Arlington, VA  22202

Chris Rouff  
Goddard Space Flight Center  
Code 522  
Greenbelt, MD  20771

Alan Perunovich  
Computer Sciences Corp.  
1100 West Street  
Laurel, MD  20707

Stephen Sacco  
CSDL  
555 Technology Square  
Cambridge, MA  02139

Vy Pham  
U. S. Army  
CECOM  
AMSEL-RD-C3-CC-L  
Fort Monmouth, NJ  07703

Ken Sall  
Century Computing Inc.  
1014 West Street  
Laurel, MD  20375-5000

Peter Pickard  
National Weather Service  
W/OS012 MS 12464  
1325 East-West Highway  
Silver Spring, MD  20910

Melvin Sanders  
National Weather Service  
1325 East-West Highway  
Rm. 12438  
Silver Spring, MD  20910

Debasis Ray  
Advanced Marine Enterprises, Inc.  
1725 Jefferson Davis Hwy, Suite 1300  
Arlington, VA  22202

Philip Shady  
Booz Allen & Hamilton  
7404 Executive Place  
Suite 500  
Seabrook, MD  20706

Thomas Roatsch  
JPL  
Mail Stop 168-414  
4800 Oak Grove Drive  
Pasadena, CA  91109-8099

Roger Sheldon  
Loral AeroSys  
7375 Executive Place  
Suite 400  
Seabrook, MD  20706
<table>
<thead>
<tr>
<th>Name</th>
<th>Company</th>
<th>Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sylvia Sheppard</td>
<td>Goddard Space Flight Center</td>
<td>Code 522, Greenbelt, Md 20771</td>
</tr>
<tr>
<td>Bill Stoffel</td>
<td>Goddard Space Flight Center</td>
<td>Code 513.1, Greenbelt, MD 20771</td>
</tr>
<tr>
<td>Elvira Shieh</td>
<td>Stanford Telecom Inc.</td>
<td>7501 Forbes Blvd. #105, Seabrook, MD 20706</td>
</tr>
<tr>
<td>Wendi Sugarman</td>
<td>Century Computing Inc.</td>
<td>1014 West Street, Laurel, MD 20707</td>
</tr>
<tr>
<td>Greg Shirah</td>
<td>Goddard Space Flight Center</td>
<td>Code 522, Greenbelt, MD 20771</td>
</tr>
<tr>
<td>Marti Szczur</td>
<td>Goddard Space Flight Center</td>
<td>Code 522, Greenbelt, MD 20771</td>
</tr>
<tr>
<td>Sharad Singh</td>
<td>Infotech Enterprises, Inc.</td>
<td>2361 Jefferson Davis Highway #1004, Arlington, VA 22202</td>
</tr>
<tr>
<td>David Taylor</td>
<td>Century Computing Inc.</td>
<td>1014 West Street, Laurel, MD 20707</td>
</tr>
<tr>
<td>Arun Jit Singh</td>
<td>Computer Sciences Corp.</td>
<td>4600 Powder Mill Road, Beltsville, Md 20705</td>
</tr>
<tr>
<td>David M. Taylor</td>
<td></td>
<td>971 Lloyd Court, Beavercreek, OH 45434-6437</td>
</tr>
<tr>
<td>Peggy C. Smith</td>
<td>Naval Surface Warfare Center</td>
<td>K52 P.O. Box 94, Dahlgren, VA 22448</td>
</tr>
<tr>
<td>My-Hanh Trinh</td>
<td>Naval Surface Warfare Center</td>
<td>10901 New Hampshire Avenue, Code B40, Silver Spring, MD 20903-5640</td>
</tr>
<tr>
<td>Gary Smith</td>
<td>Naval Surface Warfare Center</td>
<td>Code 214, Carderock Division, Bethesda, MD 20084</td>
</tr>
<tr>
<td>Walt Truszkowski</td>
<td>Goddard Space Flight Center</td>
<td>Code 522, Greenbelt, MD 20771</td>
</tr>
</tbody>
</table>
Peter Wu
Goddard Space Flight Center/SSAI
Code 912
Greenbelt, MD 20771

Jine-Hwa Yang
Computer Sciences Corp.
4600 Power Mill Road
Beltsville, MD 20705

Arleen Yeager
Goddard Space Flight Center/RMS
Code 522
Greenbelt, MD 20771

Kim Youngblood
Atlantic Research Corp.
P.O. Box 719
Fort Huachuca, AZ 85613

Ming-Hsuan Yu
Computer Science Corp.
4600 Powder Mill Road
Beltsville, MD 20705

Elissa Zizzi
NSWC DD
Dahlgren Road (K51)
Dahlgren, VA 22448
| 4. TITLE AND SUBTITLE | Transportable Applications Environment (TAE)  
|                      | Tenth Users’ Conference |
| 6. AUTHOR(S)        | Edited by Chris Rouff, Elfrieda Harris, and Arleen Yeager |
| 7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) | Goddard Space Flight Center  
|                      | Greenbelt, Maryland 20771 |
| 9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) | National Aeronautics and Space Administration  
|                      | Washington, D.C. 20546-0001 |
| 11. SUPPLEMENTARY NOTES | Elfrieda Harris and Arleen Yeager: RMS Technologies, Inc., Lanham, Maryland. |
| 12a. DISTRIBUTION/AVAILABILITY STATEMENT | Unclassified-Unlimited  
|                      | Subject Category 61  
|                      | Report available from the NASA Center for AeroSpace Information, 800 Elkridge Landing Road, Linthicum Heights, MD 21090; (301) 621-0390. |
| 13. ABSTRACT (Maximum 200 words) | Goddard Space Flight Center sponsored the Tenth TAE Users’ Conference on June 14–17, 1993. This document represents the proceedings; the papers included are published as received from the authors. This was the first TAE conference that followed an informal workshop format with panel discussions, demonstrations, tutorials and working sessions. With the TAE software being transferred into the private sector, all future user conferences will be managed by Century Computing, Inc., the commercial developers/distributors of TAE Plus. |
| 14. SUBJECT TERMS | Software, User Interface, Transportable Applications Environment |
| 15. NUMBER OF PAGES | 233 |
| 16. PRICE CODE | Unlimited |