CUSTOMIZING GRAPHICAL USER INTERFACE TECHNOLOGY FOR SPACECRAFT CONTROL CENTERS

Edward Beach and Peter Giancola
Computer Sciences Corporation
Laurel, Maryland, U.S.A

Steven Gibson
Integral Systems, Inc.
Lanham, Maryland, U.S.A

Ronald Mahmot
NASA Goddard Space Flight Center
Greenbelt, Maryland, U.S.A

ABSTRACT

The Transportable Payload Operations Control Center (TPOCC) project is applying the latest in graphical user interface technology to the spacecraft control center environment. This project of the Mission Operations Division's (MOD) Control Center Systems Branch (CCSB) at NASA Goddard Space Flight Center (GSFC) has developed an architecture for control centers which makes use of a distributed processing approach and the latest in Unix workstation technology. The TPOCC project is committed to following industry standards and using commercial off-the-shelf (COTS) hardware and software components wherever possible to reduce development costs and to improve operational support. TPOCC's most successful use of commercial software products and standards has been in the development of its graphical user interface. This paper describes TPOCC's successful use and customization of four separate layers of commercial software products to create a flexible and powerful user interface that is uniquely suited to spacecraft monitoring and control.

1. BACKGROUND

The TPOCC project was initiated in 1985 by the Control Center Systems Branch as a research and development effort. The goal was to establish an architecture that would enhance the capabilities of spacecraft control centers while simultaneously reducing the per mission cost of implementation. Through a series of prototypes, an architecture was chosen based on the following elements:

- a distributed processing approach
- industry standards
- COTS hardware and software components
- reusable custom software

The first operational system developed by TPOCC was initiated in 1989 as a replacement for the existing control center supporting the International Cometary Explorer (ICE) and the International Monitoring Platform (IMP). TPOCC successfully supported its first launch of a new spacecraft with the deployment of the Solar Anomalous and Magnetospheric Particle Explorer (SAMPEX) in July 1992. SAMPEX's users represent a special challenge for the TPOCC user interface because no SAMPEX spacecraft contact lasts more than 10 minutes. These users need the ability to rapidly access all available data, automate their activities, and quickly respond to contingencies. Other new missions have also adopted TPOCC's architecture and software as shown below.

ICE	International Cometary Explorer
IMP	International Monitoring Platform
ISTP Series	Polar Plasma Laboratory (POLAR); Solar and Heliospheric Observatory (SOHO); Interplanetary Physics Laboratory (WIND)
SMEX Series	Fast Auroral Snapshot (FAST); Solar Anomalous & Magnetospheric Particle Explorer (SAMPEX); Submillimeter Wave Astronomy (SWAS)
TRMM	Tropical Rainfall Measurement Mission
XTE	X-Ray Timing Explorer

Satellites Employing the TPOCC Architecture

TPOCC's commitment to a standards-based approach led to the selection of the X window system from the Massachusetts Institute of Technology (MIT) as the core of the TPOCC display subsystem. Three other layers of commercial software complement the X protocol and complete the TPOCC user interface capabilities. The complete history of the TPOCC display subsystem is shown below, starting with the early work evaluating various windowing systems (X, NEWS, and SunView) and leading up to the present design that includes COTS products such as Hewlett Packard's (HP's) Visual User Environment (VUE) and the user interface design tool Builder Xcessory. All of the milestones listed in the table are explained in detail in the sections that follow.

1988	NEWS evaluated for possible use Prototype built using SunView
1989	Use of X Windows initiated (X11R3) Prototype using Athena widget set
1990	Prototype ported to Motif widget set UIL becomes display definition language
1991	TPOCC page editor written Port to X11R4 and Motif 1.1.3 Graphical widgets added (dials and plots)
1992	HP VUE integrated SAMPEX launched Command panel written Builder Xcessory integrated

History of the TPOCC Display Subsystem

The overriding philosophy behind TPOCC's display design is that close adherence to the leading open systems standards will provide tremendous benefits. The importance of this tenet is highlighted in the sections that follow. The standard products selected by TPOCC have many significant features, such as:

- hardware independence
- remote display distribution
- support for object-oriented software design
- extensible widget set
- extensible display definition language

But in addition to these features, a strict policy of standards adherence provides benefits of its own. It allows rapid integration of COTS products based on the same standards. It has also made it possible for TPOCC to jointly develop a common library of display objects with projects in different NASA organizations. Finally, it allows external organizations to create graphical applications that fit into the TPOCC user environment with little or no modification.

2. THE X WINDOW SYSTEM

The distributed environment used by the TPOCC architecture necessitates the use of a network based windowing system. The X Window System was chosen to serve as the basis for TPOCC's graphical user interface because of its wide acceptance and superior performance. The other alternatives available at the time (late 1988) were either slow and not widely accepted (NEWS), or vendor-specific (such as SunView). In 1989, TPOCC made a full-scale commitment to X and began using the software shipped with MIT's X Version 11 Release 3 as the lowest layer supporting the rest of TPOCC's user interface capabilities.

X Window's client/server model layered on networking standards such as Ethernet and Transmission Control Protocol (TCP)/ Internet Protocol (IP) is identical to the distributed processing techniques used throughout the TPOCC architecture. The benefits of this client/server model include:

1. <u>hardware independence</u>: any terminal supporting the X protocol can display data, regardless of graphics hardware

2. <u>remote display distribution</u>: displays can be sent to remote locations on the Ethernet without modifying software

3. <u>reduced cost per user position</u>: the ability to use X-terminals reduces the cost of a single user position to less than half the cost of a workstation

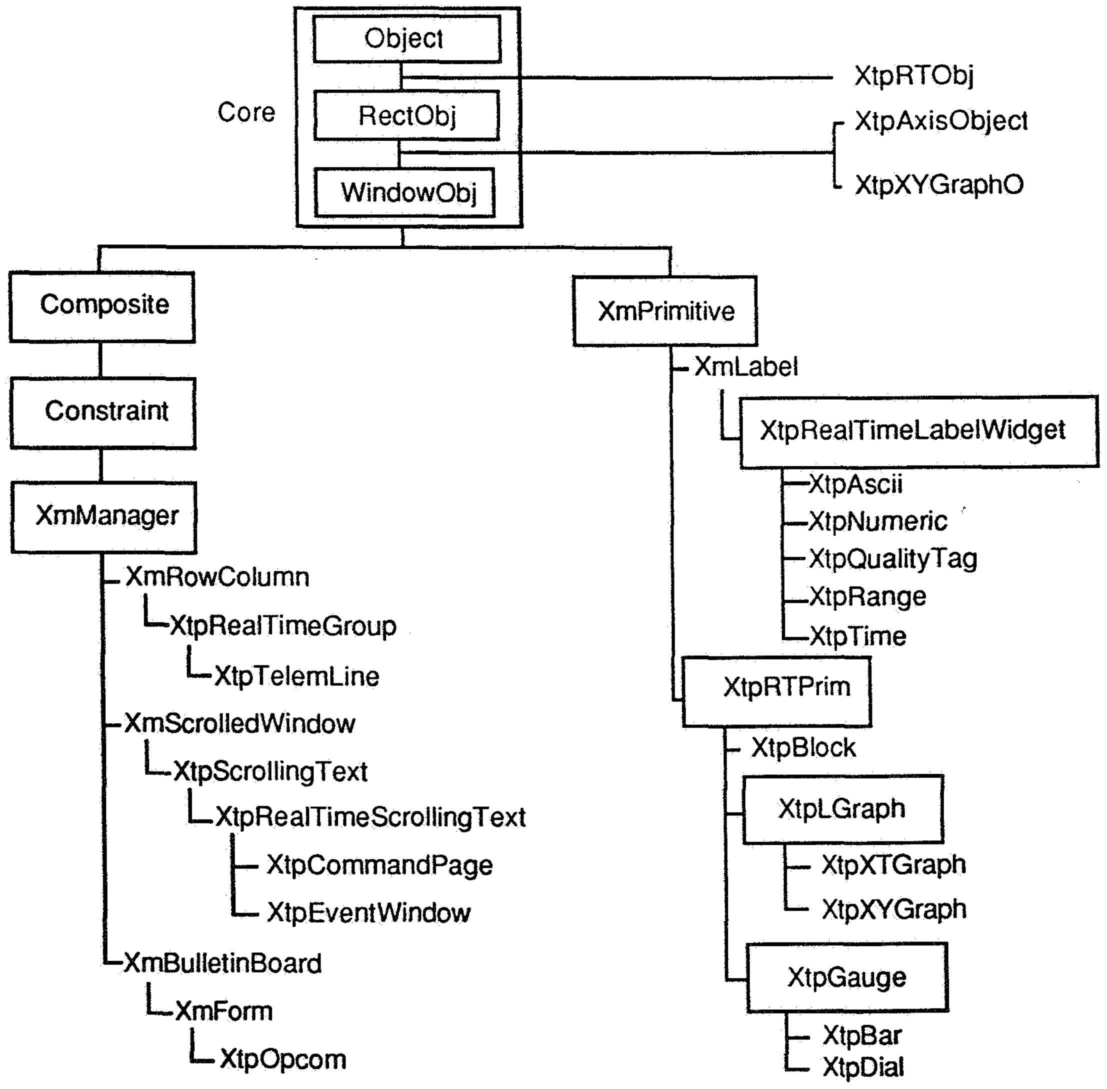
MIT supplies the "X lib" software library to support the client side of this model and to provide routines that can be called to build proper X protocol messages. However, another library layer exists in X, namely the X toolkit. This collection of routines support an object-oriented paradigm in a traditional programming language (C). The toolkit allows the creation of sets of display objects, called widgets, that exhibit object-oriented properties such as inheritance, data hiding, and polymorphism. TPOCC has taken advantage of the power of the toolkit to create their own custom widget set that is tailored to a real-time control center environment. Furthermore, any widget written according to the X toolkit standard can be easily shared by another project following the same approach. TPOCC is currently exchanging widgets with several projects including Goddard's Data Systems Technology Division's Generic Spacecraft

Analyst Assistant (GenSAA) project. In effect, the projects are creating a common pool of display capabilities that can be developed in pieces but rapidly merged when an operational system is being put together.

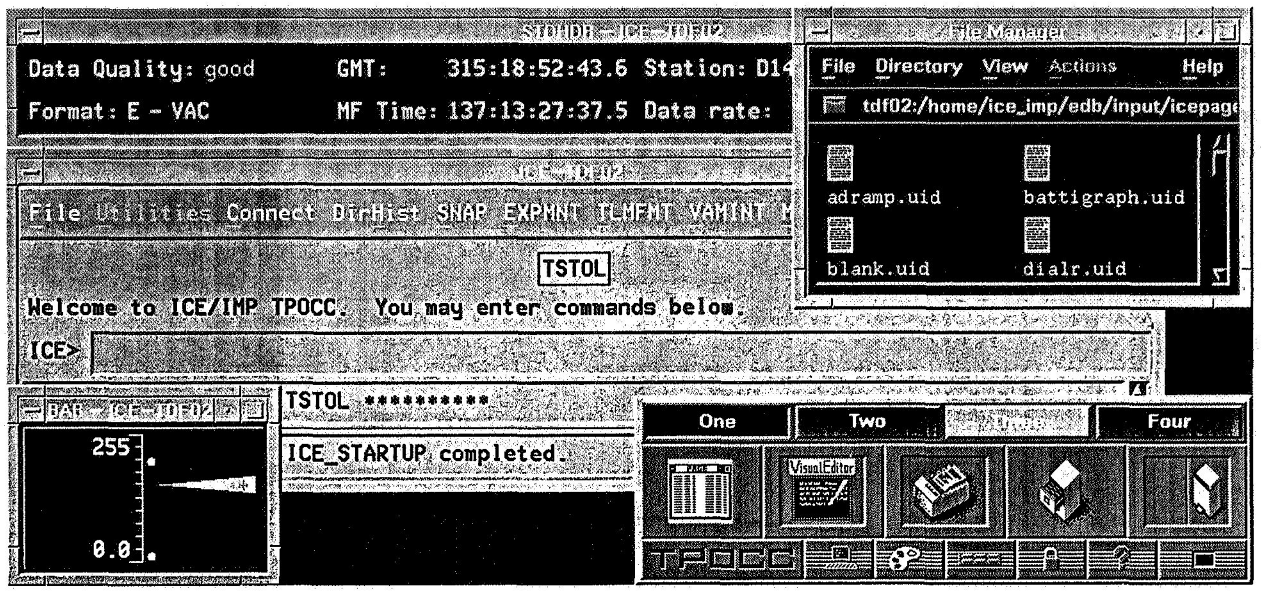
3. OSF/MOTIF

TPOCC's earliest widget set was an extension of the Athena widget set supplied with MIT's X window software distributions. The publically-available Athena widgets were used as containers to hold individual display objects and as labels and annotations on TPOCC's displays. The TPOCC widget set inherited properties of some of the existing Athena widgets to simplify coding. However, the Athena widget set had numerous shortcomings at the time, principally because it had been written more as an example of X toolkit coding techniques than as a viable commercial product. TPOCC began to look elsewhere for a more robust widget set, preferably the one emerging as the industry standard. Selecting a widely used widget set would ensure that all TPOCC applications exhibited a standard "look and feel" that would be consistent with any commercial software tools that could be integrated into the TPOCC environment.

So, the Open Software Foundation's (OSF's) Motif user environment was selected, because it enjoyed the widest support in the Unix community and for its extensible user interface language (UIL) discussed below. TPOCC's widget set was ported to Motif in 1990 and now, in 1992, includes over 20 widgets (identified by TPOCC's "Xtp" prefix below).



TPOCC Widget Hierarchy


File Directory View Actions Help
File Utilities Connect DirHist SNAP EXPMNT TLMFMT VAMINT
TSTOL
255
0.0
One
Two
Four
VisualEditor
TPOCC

```
Data Quality: good      GMT:     315:18:52:43.6  Station: D14
Format: E - VAC         MF Time: 137:13:27:37.5  Data rate:
```

```
tdf02:/home/ice_imp/edb/input/icepage
adramp.uid        battigraph.uid
blank.uid         dialr.uid
```

```
Welcome to ICE/IMP TPOCC. You may enter commands below.
ICE>
```

```
TSTOL **********
ICE_STARTUP completed.
```

TPOCC's HP VUE Window Environment

a fully graphical interface that does not require the knowledge of any low-level command languages (such as the Unix shells and UIL). In addition, the TPOCC project has customized HP VUE to translate many control center operations into graphical sequences involving buttons, menus, and icons. As an example, all TPOCC applications are invoked through VUE icon selections. Similarly, files can be edited, printed (in various styles according to the POCC file type), or deleted just by dragging their icon to the editor, printer, or trash can drop target.

Perhaps the most powerful feature of HP VUE is the concept of workspaces. A workspace is a logical grouping of windows that allows the user to view just a single group at a given time. Once each window has been brought up the user can assign it to any workspace through a simple menu operation. Switching between workspaces is as simple as pushing the proper button on the HP VUE workspace manager panel (located at the bottom right of a TPOCC display as shown on the screen above). This capability proved to be especially useful in support of the SAMPEX launch. During launch and early orbit checkout, up to 32 users were using the TPOCC system, with each user viewing from 5 to 10 different pages. Since there is far more information on this number of pages that can be viewed on a single console, the ability to group the windows proved invaluable. Each user would define four workspaces and would cycle through to each group as necessary.

6. FUTURE DIRECTIONS

Although TPOCC's graphical user interface has come a long way in the last four years, active work continues. New widgets to support schematic displays are being built in conjunction with GenSAA. The widget set is being modified to support data from any arbitrary source, not just the TPOCC data server process. Motif's new support for drag-and-drop operations will be incorporated. Several new applications are being developed, including definition file editors and a graphical debugger for the TPOCC System Test and Operations Language (TSTOL). Improved support for TSTOL itself enhances the user interface, for TSTOL complements the display software by allowing sequences of user activities (such as page selection and placement) to be automated through a set of commands known as a TSTOL procedure. Other capabilities will be added to the reusable software base as requirements are received from new missions. However, the goals established when the TPOCC display development began appear to have been met. A powerful, efficient user interface that improves operational capabilities has been created at minimal cost by maximizing the use of commercial software. Just as importantly, the TPOCC display subsystem has achieved complete reusability from one mission to the next. The display software is one of TPOCC's few subsystems that does not require mission-unique code to support different spacecraft. Each mission defines unique displays in UIL, but the costly process of creating or modifying software for each mission is eliminated.

7. REFERENCES

1. Mahmot, Ronald and Beach, Edward. 1992. TPOCC Graphical User Interface for Editing Display Pages. *1991 Research and Technology Report, Goddard Space Flight Center*, 208-211.

```
! @(#) File name: dial.uil  Release: 1.9
module dial_page
names = case_sensitive
include file 'XmAppl.uil';
include file 'XtpAppl.uil';
value line_offset: 20;
object
   XtpTopLevel: XmForm
     { arguments
       {
       };
       controls
       {
         XmLabel page_title;
         XtpDial dial1;
       };
     };
   page_title: XmLabel
     { arguments
       {
         XmNtopAttachment = XmATTACH_FORM;
         XmNleftAttachment = XmATTACH_FORM;
         XmNrightAttachment = XmATTACH_FORM;
         XmNleftOffset = 5;
         XmNrightOffset = 5;
         XmNlabelString = compound_string(
              '- XtpDial Widget Example -',separate=true)
              & '(absolute resize)';
       };
     };
   dial1 : XtpDial
     { arguments
       {
        ! XmForm Constraint resources
          XmNtopAttachment = XmATTACH_WIDGET;
          XmNtopWidget = page_title;
          XmNtopOffset = line_offset;
          XmNleftAttachment = XmATTACH_FORM;
          XmNrightAttachment = XmATTACH_FORM;
          XmNbottomAttachment = XmATTACH_FORM;
        ! XtpDial resources - XmPrimitive Class Part
          XmNborderWidth = 2;
        ! XtpDial resources - XtpRTPrim Class Part
          XtpNmnemonic = default_cvt_uli1;
          XtpNprocess = default_decom;
          XtpNvariableType = XtpCVT_ULI;
        ! XtpDial resources - XtpGauge Class Part
          XtpNscaleMax = "255";
          XtpNshowMinmax = true;
        ! XtpDial resources - XtpDial Class Part
          XmNmargin = 4;
          XtpNdrawPost = true;
       };
     };
end module;
```

UIL Definition for TPOCC Page "DIAL"

UIL is a powerful language that supports all possible combinations of Motif-based widgets. However, one lesson learned during the TPOCC effort has been that end user creation of display windows directly in UIL is too time consuming and requires detailed knowledge of the Motif widget set. With this problem in mind, the TPOCC page editor was created. This application, written to comply with Motif's Style Guide, allows users to rapidly design pages using a row-column layout. About eighty percent of control center telemetry displays have been traditionally formatted this way. The intuitive Motif interface provided by the page editor has reduced the time required to construct a page from several hours to about twenty minutes (Ref. 1).

4. BUILDER XCESSORY

However, the page editor does not solve the problem of laying out displays with merged text and graphics. The graphical elements of the TPOCC widget set necessitate a drag-and-drop style layout and design tool. TPOCC therefore began a search for a user interface management system (UIMS) that allows user created widgets to be incorporated into the tool. Builder Xcessory (BX) from Integrated Computer Solutions is a UIMS that uses UIL's WML facility to define new widgets for the BX palette. Thus, TPOCC can create a single WML definition file and come up with both an extended UIL compiler and an extended design tool. TPOCC is in the final phase of integration with Builder Xcessory and expects to be installing this product in the operational TPOCC control centers in early 1993.

Although the primary purpose for integrating Builder Xcessory is to allow users to develop more complex pages than those supported by the page editor, this tool is also quite useful to TPOCC software developers. In this domain, BX is used to prototype screen designs for new Motif applications. Before writing any code, TPOCC developers demonstrate the BX "mock-up" of the future application to gain valuable ideas and feedback from TPOCC users. The command panel, an application that lets flight operations team members plan and conduct the activities for each spacecraft contact through a simple button interface, was prototyped in this manner. Two rounds of demonstrations were held before a final design was reached. The early feedback possible with such a methodology helps reduce the life cycle costs of developing this type of application; prototyping is now a standard part of developing a new application for TPOCC.

5. HP VUE

Finally, TPOCC is using Hewlett Packard's Visual User Environment (HP VUE), the fourth and final layer of COTS software, to provide a graphical interface to the Unix operating system. Shielding the user from Unix's command language is a significant step toward TPOCC's goal of creating

3.1 TPOCC Widget Set

The real-time telemetry display widgets supported by TPOCC can be divided into three types, with each type supporting a different form of data representation. A fourth type, schematic widgets, will be added in the near future and is discussed in the final section below.

3.1.1 Textual Widgets

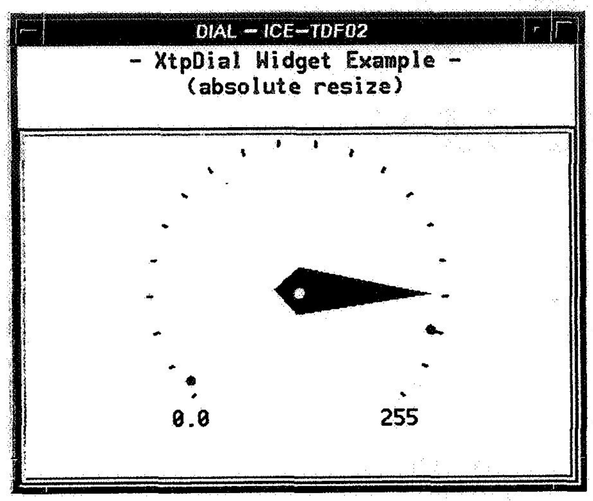
The textual widgets in the TPOCC widget set are divided into two types: those based on Motif's Label widget and those based on Motif's List widget. TPOCC's label widgets are used to animate different types of real-time data at a fixed screen location. Widgets of this type include XtpAscii, XtpNumeric, XtpRange, and XtpTime. TPOCC's list widgets are used to animate a scrolling display of lines of text. Widgets of this type include XtpCpage and XtpEvent.

3.1.2 Gauge Widgets

The first graphical widgets implemented for the TPOCC widget library were gauge widgets. Currently, two TPOCC gauges exist, XtpBar and XtpDial. An XtpDial widget is shown at right. The gauge widgets animate based on a single telemetry point and show not only the current value of the parameter but also it's minimum and maximum values during the current spacecraft contact.

3.1.3 Plot Widgets

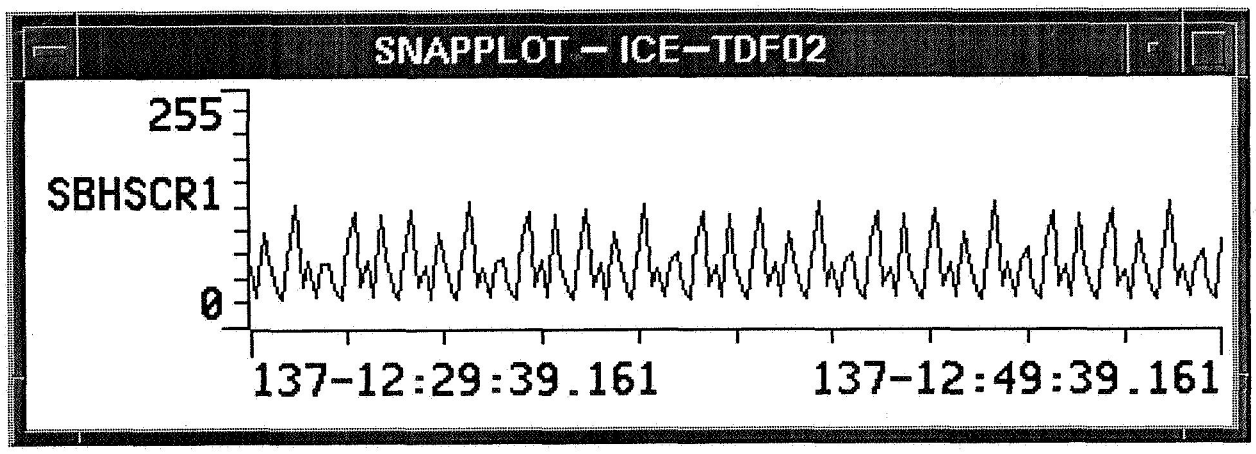
The most sophisticated widgets are the plot widgets. TPOCC currently supports two such widgets: XtpXTGraph and XtpXYGraph. XtpXTGraph stripcharts a single parameter against time and scrolls based upon a user-selected policy. XtpXYGraph can plot up to five parameters against either a sixth parameter or time. Two independent Y scales are supported, but the X axis is fixed (scrolling is not supported). An XtpXYGraph widget is shown receiving real-time data below.



TPOCC's Dial Widget

3.2 TPOCC's Use of UIL

Motif's UIL plays a key role in TPOCC's display design. Not only does UIL provide the ability to augment the Motif widget set through OSF's Widget Meta Language (WML), but it also allows unlimited flexibility in page design and pages to be created without rebuilding any software. Through WML, the UIL language can be expanded to include user-defined widgets such as the real-time set developed by TPOCC. When WML is run, a new UIL compiler is generated that accepts specifications for TPOCC widgets as valid syntax. A TPOCC UIL fragment is shown on the following page. This UIL defines a single control center display window, known as a page. The definition is for the TPOCC Dial widget window shown in the figure above.



TPOCC's X-Y Graph Widget