LA BOÎTE À OUTILS

OSF/MOTIF 1.1

Vincent BOUTHORS / Philippe KAPLAN
Bull Research, Sophia Antipolis

BUT DU COURS

CONNAISSANCE

- des fonctions des base
- de l'ensemble des classes
- des attributs les plus utiles
- des concepts de traitement des événements

RÉALISATION

- d'exemples simples
- d'un éditeur de graphe simplifié

PLAN GÉNÉRAL

Cours OSF/MOTIF -- ILOG & BULL -- droits de reproduction réservés

Plan du cours

1er jour
- introduction
- documentation
- classes élémentaires *

2ème jour
- primitives
- géométrie *
- menus *

3ème jour
- événements
- classes élaborées *
- shell *

(* = travaux pratiques)

Cours OSF/MOTIF -- ILOG & BULL -- droits de reproduction réservés

Historique des systèmes de fenêtres

XEROX : Alto, Cedar, Star

APPLE : Lisa, Mac-Intosh

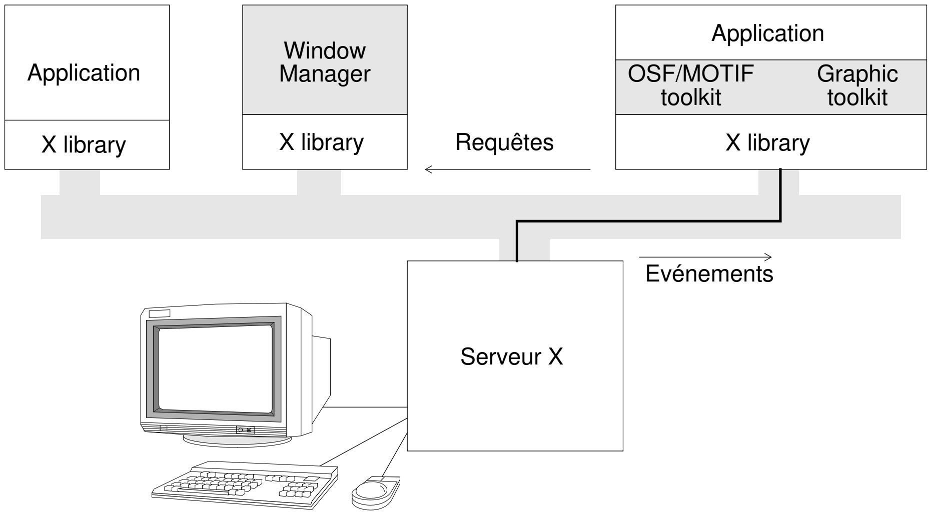
divers : W, Fenix, Andrew, Vitrail, Dialog, MS-windows

SUN : NEWS

CONSORTIUM (MIT) : X-window

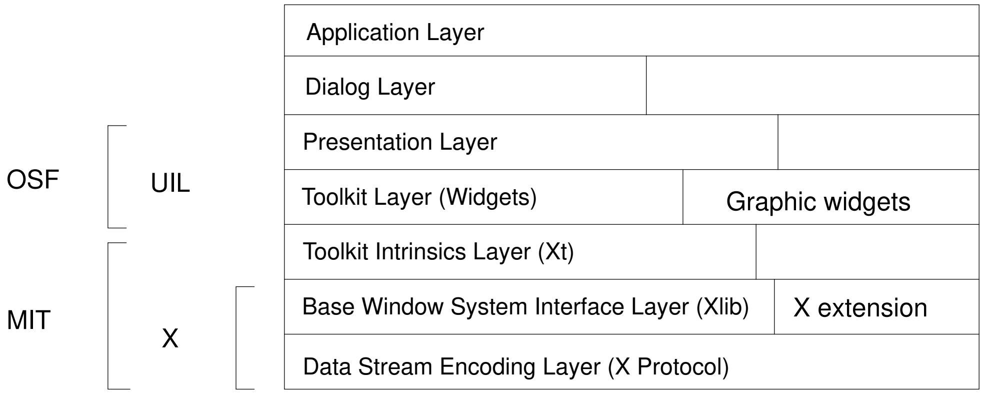
Introduction

Architecture client/serveur de X

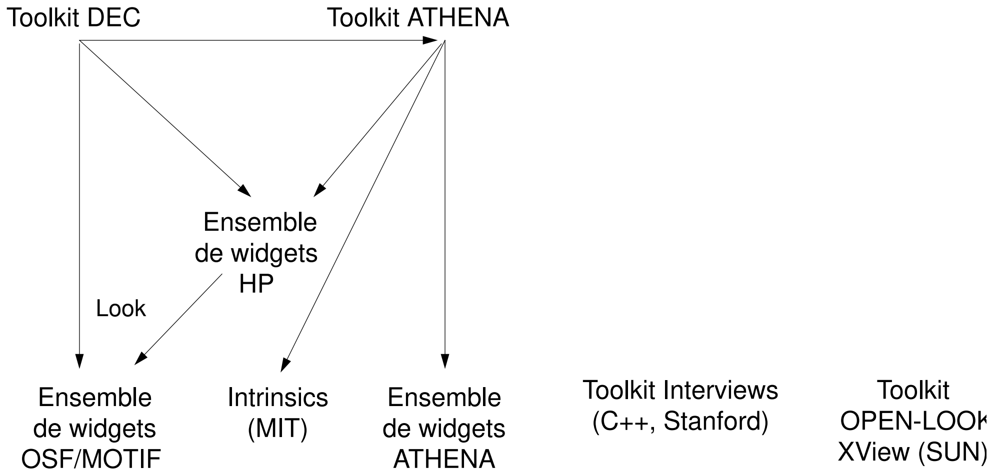


Historique

Couches logicielles (2)



Historique des toolkits sur X



Introduction à la toolkit Motif

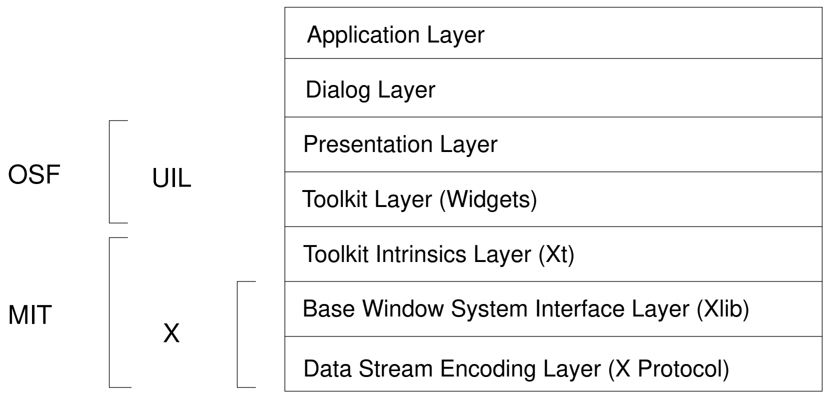
Définition d'un widget

Facettes d'un widget

Approche Orientée Objet

Principales primitives

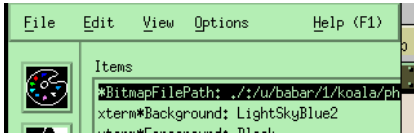
Couches logicielles (1)

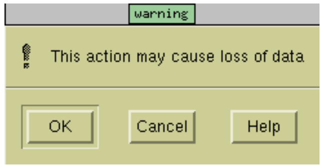


Widget

Window Object

"machin"





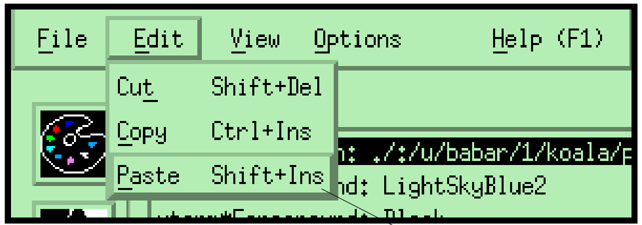
Facettes d'un widget

Apparence

Géométrie

Comportement

Sémantique



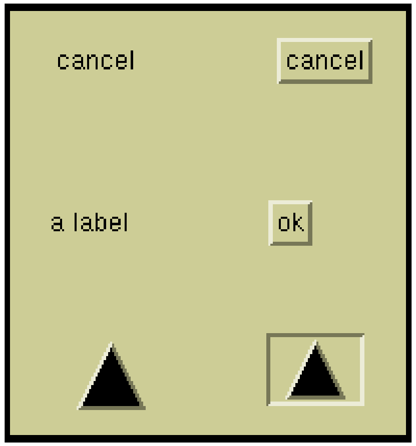

```
PasteCallback(...)
{
    ...
}
```


Approche orientée objet : Classes et Objets

Classes:

Objets:

(instance, occurence)



Approche orientée objet : héritage

Héritage :

super classe	*Label*
↑	↑
sous-classe	*PushButton*

Attributs (champs) :

- Label : *background, foreground, labelString*
- PushButton : *background, foreground, labelString, activateCallback*

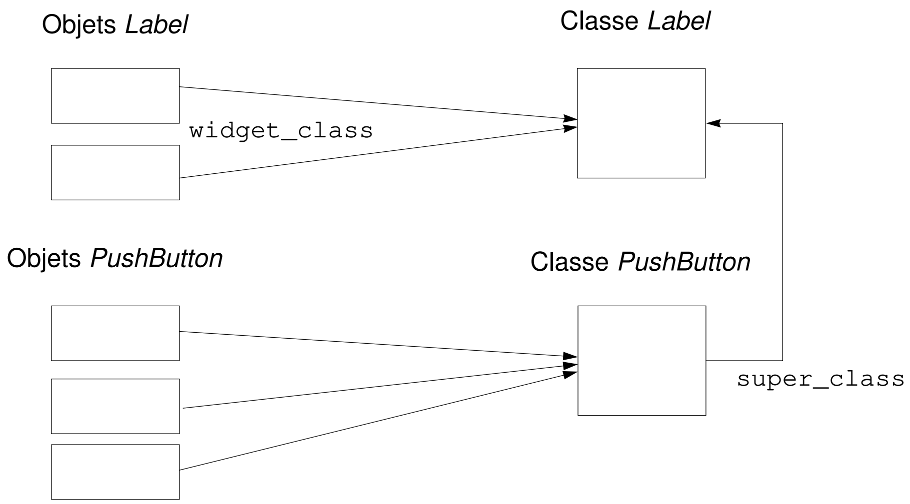
Méthodes (fonctions):

- Label : Redraw => affichage de labelString
- PushButton : Redraw => Redraw de Label, puis dessin du relief

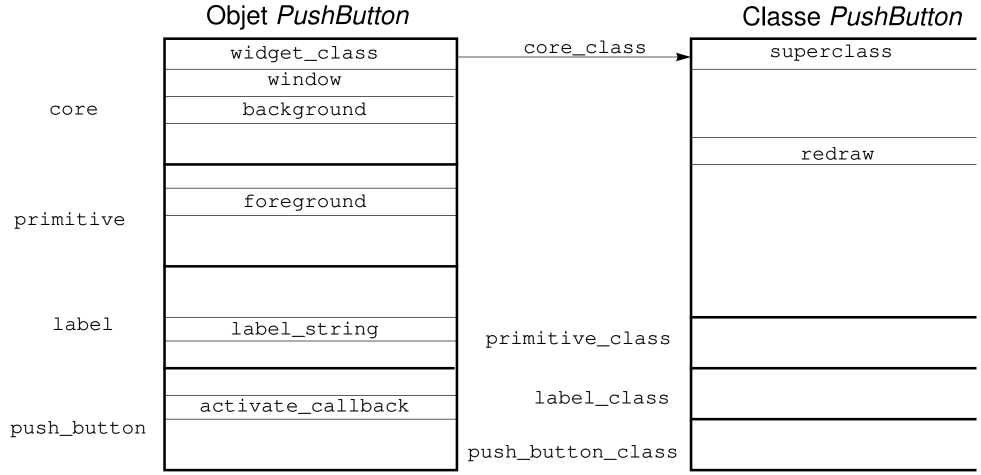
Surcharge (overloading) : même nom pour des fonctions différentes

Redraw(label) Redraw(pushButton) Redraw(object)

Toolkit: Noyau Orienté Objet en C



Toolkit: noyau orienté objet en C




```
widget->core.widget_class->core_class.redraw(widget);
```


Principales primitives de la toolkit : Création

XtCreateWidget

widget= XtCreateWidget(name, class, parent, args, numArgs)

```
Widget box, button1;
Arg args[10];
int n;

n = 0;
XtSetArg(args[n], XmNx, 50); n++;
XtSetArg(args[n], XmNy, 20); n++;
```

attribut name	valeur value
XtNx	50
XtNy	20

n

```
button1= XtCreateWidget("quit", xmPushButtonWidgetClass, box, args, n);
```


Principales primitives de la toolkit : Modification

XtSetValues

XtSetValues(widget, args, numArgs)

```
Widget scale;
Arg args[10];
int n;

n = 0;
XtSetArg(args[n], XmNvalue, 17); n++;
```

attribut name	valeur value
XmNvalue	17

n

```
XtSetValues(scale, args, n);
```

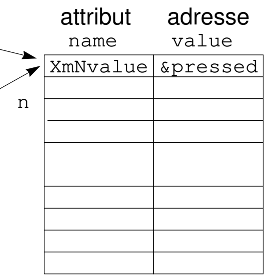
Principales primitives de la toolkit : Consultation

XtGetValues

XtGetValues(widget, args, numArgs)

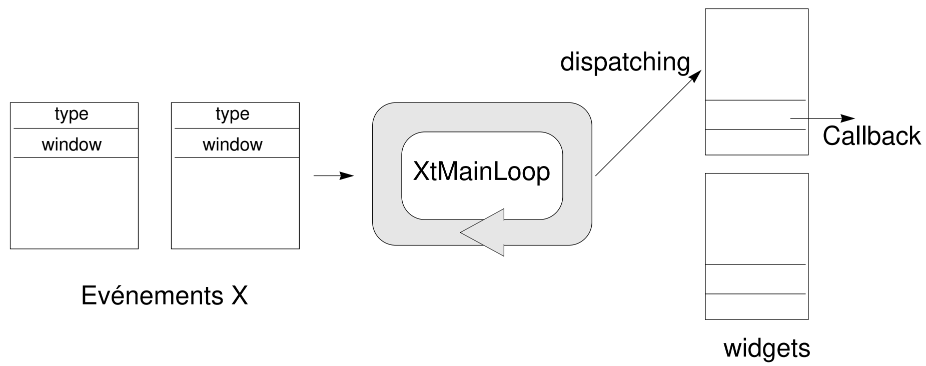
```
Widget scale;
Arg args[10];
int n;
Boolean pressed;

n = 0;
XtSetArg(args[n], XmNset, &pressed); n++;
```




```
XtGetValues(scale, args, n);
```


Principales primitives de la toolkit : Boucle principale



Premier exemple

Description de l'exemple

Hiérarchie des widgets

Initialisation de la toolkit

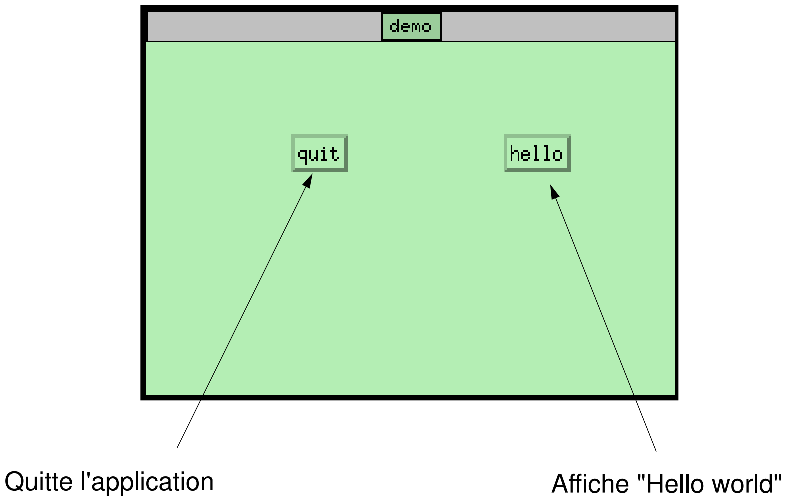
Ressources

Mise en œuvre des widgets et des callbacks

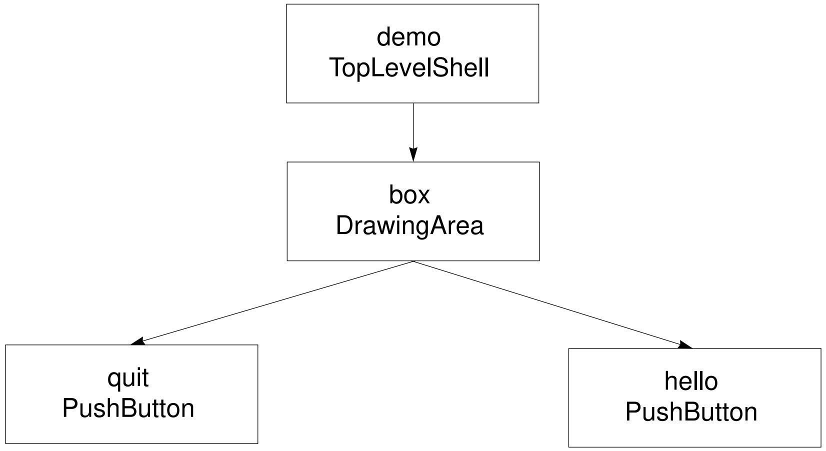
Organisation du programme

Listing

Premier exemple : description



Premier exemple : Hiérarchie des widgets



Initialisation de la toolkit

widget= XtInitialize(shellName, applicationClass, options, numOptions, argc, argv)

```
topLevel= XtInitialize("demo",  "Demo", NULL, 0, &argc, argv);
```

Noms pour les ressources

Options supplémentaires
par défaut : -fg <color>
-bg <color>
-iconic
...

Arguments de la ligne de commande

Ressources

par les noms :

attribut

```
demo.box.quit.labelString : bye
```

chemin (path) valeur

par les classes :

```
Demo.XmDrawingArea.XmPushButton.Foreground: yellow
```

mixte :

```
Demo*XmPushButton.background:grey70
```

se rappeler des règles de priorités:

```
*background:        grey70
*quit.background:   red
```


Mise en œuvre des widgets

Création:

```
XtCreateWidget(name, class, father, args, numArg);
```

pour créer l'objet en mémoire

Management:

```
XtManageChild(widget);
```

pour qu'il soit pris en compte par son père

Réalisation:

```
XtRealizeWidget(topLevel);
```

pour créer la fenêtre X associée

S'applique récursivement aux fils, automatique si le père est déja réalis

Mise en œuvre des Callbacks

Définition de la callback

```
void PrintCallback(widget, clientData, callData)
     Widget widget;
     XtPointer clientData;
     XtPointer callData;
{
     char *text= (char *) clientData;
     fprintf(stderr, "%s\n", text);
}
```

Enregistrement de la callback

```
XtAddCallback(helloButton, XmNactivateCallback,
              PrintCallback, "Hello World");

XtAddCallback(quitButton, XmNactivateCallback,
              PrintCallback, "That's all folks");
```


Organisation du programme

- Partie présentation
  - Initialisation de la toolkit
  - Création des widgets et enregistrement des callbacks
- Partie dialogue
  - Boucle infinie de traitement des événements
  - Callbacks
  - Variables statiques

Listing de demo.c (1)


```
/*****************************************************************************************\
                         MOTIF Demonstration Program
\*****************************************************************************************/
#include <stdio.h>
#include "X11/Intrinsic.h"
#include "Xm/Xm.h"
#include "Xm/DrawingA.h"
#include "Xm/PushB.h"
/*=======================================================================*\
                          Dialog part: Callbacks
\*=======================================================================*/
void QuitCallback(widget, clientData, callData)
    Widget widget;/* unused */
    XtPointer clientData;/* unused */
    XtPointer callData;/* unused */
{
    fprintf(stderr, "Bye\n");
    exit(0);
}
void PrintCallback(widget, clientData, callData)
    Widget widget;/* unused */
    XtPointer clientData;/* text */
    XtPointer callData;/* unused */
{
    char * text = (char *) clientData;
    fprintf(stderr, "%s\n", text);
 }
```


Listing de demo.c (2)


```
/*=======================================================================*\
                       Presentation part: Creation
/*=======================================================================*\

Widget topLevel, box, widget;
Arg args[10];
int n;
void PresentationMake(argcPtr, argv)
    int * argcPtr;
    char ** argv;
{
topLevel = XtInitialize("demo1", "Demo", NULL, 0, argcPtr, argv);
n = 0;
XtSetArg(args[n], XmNwidth, 300); n++;
XtSetArg(args[n], XmNheight, 200); n++;
XtSetArg(args[n], XmNtraversalOn, True); n++;
XtSetArg(args[n], XmNresizePolicy, XmRESIZE_GROW); n++;
box = XtCreateWidget("box", xmDrawingAreaWidgetClass, topLevel, args, n);
XtManageChild(box);
n = 0;
XtSetArg(args[n], XmNx, 80); n++;
XtSetArg(args[n], XmNy, 50); n++;
widget= XtCreateWidget("quit", xmPushButtonWidgetClass, box, args, n);
XtManageChild(widget);
XtAddCallback(widget, XmNactivateCallback, PrintCallback, "That's all folk");
XtAddCallback(widget, XmNactivateCallback, QuitCallback, NULL);

n = 0;
XtSetArg(args[n], XmNx, 200); n++;
XtSetArg(args[n], XmNy, 50); n++;
widget= XtCreateWidget("hello", xmPushButtonWidgetClass, box, args, n);
XtManageChild(widget);
XtAddCallback(widget, XmNactivateCallback, PrintCallback, "Hello world");

XtRealizeWidget(topLevel);
}
```

Listing de demo.c (3)


```
/*=====================================================================*\
                              Program
\*=====================================================================*/


main(argc, argv)
    int argc;
    char ** argv;
{
    PresentationMake(&argc, argv);

    XtMainLoop();
}


\*=====================================================================*/
```


La boîte à outils "Intrinsics"

- Introduction
  - Historique
  - Introduction
  - Premier exemple
- la boîte à outils "Intrinsics"
  - Documentation
  - Primitives
  - Traitement des événements
- Classes
  - Elémentaires
  - Concernant la Géométrie
  - Menus
  - Elaborées
  - Shell
- Hiérarchie des widgets

Documentation

- User Guide
- Style Guide
- Programmer's Guide
  - Widgets
  - MWM
  - UIL
  - Intrinsics
- Programmer's Reference
  - manual
  - AES

Documentation: Programmer's reference manual

XmToggleButton

include: Xm/ToggleB.h

xmToggleButtonWidgetClass

Ressources: nom, classe, type, valeur par défaut, accès

```
  XmNset, XmCset, Boolean, False, CGS
```

C XtCreateWidget
G XtGetValues
S XtSetValues

Structure paramètre des callbacks

```
  typedef struct {
      int          reason;
      XEvent *event;
      int set;
  } XmToggleButtonCallbackStruct;
```

Translations

```
  BSelectPress:      Arm()
  BSelectRelease:  Select() Disarm()
  ...
```

Related information: XmCreateToggleButton

Virtual Bindings

Ils sont utilisés dans les documents pour éviter de préciser l'événement physique réel.

Boutons

BSelect	<Btn1>
BDrag, BPrimaryPaste	<Btn2>
BCustom, BMenu	<Btn3>
BPrimaryCopy	Ctrl<Btn2>
BPrimaryCut	Alt<Btn2>

Modifiers

MShift	<Shift>
MCtrl	<Ctrl>
MAlt	<Alt>

Keys

KActivate	<Return>
KHelp	<Help> ou <F1>
KMenuBar	<F10>
KNextPane	<F6>

Convention de nommage

Préfixe selon la couche

X	X-window	XOpenDisplay, XCreateWindow
Xrm	Resource Manager	XrmGetResource
Xt	X Toolkit	XtOpenDisplay, XtCreateWidget
Xm	OSF/MOTIF	XmCreatePushButton, XmTextInsert

Minuscules et Majuscules

Fonctions	XOpenDisplay
Constantes	Button1, XA_PRIMARY
variables	x, borderWidth, ou border_width

Primitives

- ApplicationContexts, Displays, TopLevels
- Ressources et Options
- Fonctions de convénience
- Map et Manage
- Consultation et Modification
- Conversion
- Copier/Coller: Séléction

A la place de XtInitialize

Initialisation de la toolkit

```
XtToolkitInitialize();
```

Création d'un ApplicationContext

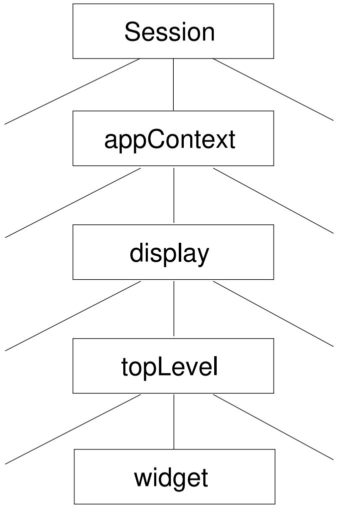
```
XtAppContext appContext= XtCreateApplicationContext();
```

Ouverture d'une connexion avec le serveur X

```
Display * display= XtOpenDisplay(appContext, displayString,
                                 applicationName, application Class,
                                 options, numOptions, argc, argv);
```

Création d'un widget shell

```
Widget topLevel= XtAppCreateShell(applicationName,
                                  applicationClass, widgetClass,
                                  display, args, numArgs);
```



Valeurs par défaut des attributs

dans le programme

XtCreateWidget(...)

dans l'environement de l'application

$XUSERFILESEARCHPATH   exemple: $HOME/%L/%N:$HOME/%N
$XAPPLRESDIR/<class>
$HOME/<language>/<class> ou $HOME/<class>

/usr/lib/X11/<language>/app-defaults/<class>
/usr/lib/X11/app-defaults/<class>

dans l'environement de l'utilisateur pour une machine précise

$XENVIRONMENT, sinon $HOME/.Xdefaults-<host>

dans l'environement de l'utilisateur pour toutes les machines

propriété RESOURCE_MANAGER (xrdb), sinon $HOME/.Xdefaults

dans la ligne de commande

... -bg grey80 ...

Options par défaut de la ligne de commande

Options	Exemples
-display display	-display babar:0.0
-geometry geometry	-geometry 230x140+10-10
-bg color, -background color	-bg Blue
-bd color, -bordercolor color	-bd red
-bw number, -borderwidth number	-bw 2
-fg color, -foreground color	-fg white
-fn font, -font font	-fn 9x15
-iconic	-iconic
-name	-name
-rv, -reverse	-rv, -reverse
+rv	+rv
-selectionTimeout	-selectionTimeout
-synchronous	-synchronous
-title string	-title demo
-xnllanguage language[_territory][.codeset]	-xnllanguage Japan
-xrm resourcestring	-xrm "*fontList: 6x13"

Options

XtOpenDisplay(appContext, displayString, applicationName, applicationClass
options, numOptions, argc, argv);

```
XrmOptionDescRec options= {
/* option        specifier          argKind              value    */
  {"-bg",        "*background",     XrmOptionSepArg,     NULL}
  {"-iconic",    ".iconic",         XrmOptionNoArg,      "true"}
  {"-xrm",       NULL,              XrmResArg,           NULL}
}
```

XrmGetRessource(dataBase, stringName, stringClass, type, value);

```
String type;                                    unsigned int    size
XrmValue value;                                 caddr_t         addr
XrmGetRessource(&diplay->db, "demo.iconic", "Demo.Iconic",
               &type, &value);
```


Nombre variable d'arguments : varargs

XtCreateWidget (name, class, parent, args, numArgs)

XtVaCreateWidget(name, class, parent, ...)

```
XtVaCreateWidget("quit", xmPushButtonWidgetClass, box,
                 XmNx, 50,
                 XmNy, 20,
                 NULL, 0);
```

Modification : exceptions à XtSetValues

XtManageWidget(widget)
XtUnmanageWidget(widget)

XtRealizeWidget(widget)
XtUnrealizeWidget(widget)

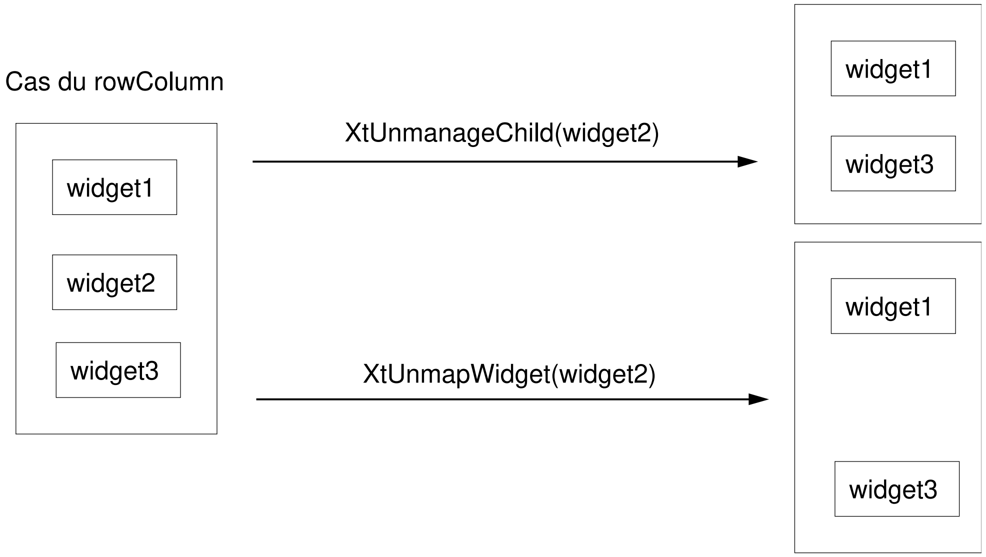
XtMapWidget(widget)
XtUnmapWidget(widget)

XtSetSensitive(widget, sensible)

XtAugmentTranslations(widget, translations)
XtOverrideTranslations(widget, translations)

XtAddCallback(widget, callbackAttribute, procedure, clientData)

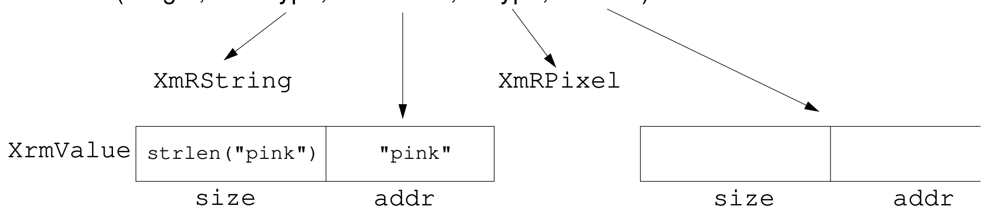
Map et Manage



Cours OSF/MOTIF -- ILOG & BULL -- droits de reproduction réservés

Conversion : cas général

XtConvert(widget, fromType, fromValue, toType, toValue)




```
XrmValue fromValue, toValue;
fromValue.size= strlen("pink") + 1;
fromValue.addr= "pink";
XtConvert(widget, XmRString, &fromValue, XmRPixel, &toValue);
if (toValue.addr != NULL) {
   pixel= * (Pixel *) toValue.addr;
}
```

Cours OSF/MOTIF -- ILOG & BULL -- droits de reproduction réservés

Consultation : exceptions à XtGetValues

XtDisplay(widget)
XtScreen(widget)
XtWindow(widget)

XtDisplayOfObject(widgetOrGadget)
XtScreenOfObject(widgetOrGadget)
XtWindowOfObject(widgetOrGadget)

XtIsSensitive(widget)

XtParent(widget)

XtName(widget)

XtIsManaged(widget)
XtIsRealized(widget)

XtClass(widget)
XtSuperClass(widget)
XtIsSubclass(widget, class)
XtIsSubclass(widget, class)

XtWindowToWidget(window)

XtNameToWidget(ancestor, path_name)

Cours OSF/MOTIF -- ILOG & BULL -- droits de reproduction réservés

Fonctions de convénience

Alternative à XtCreateWidget

XtCreateWidget (name, class, parent, arg, numArgs)

XmCreate <class> (parent, name, args, numArgs)

XmCreatePushButton

Pseudo classe

XmCreatePulldownMenu, XmCreatePopupMenu, XmCreateOptionMenu

XmCreateDialogError, XmCreateDialogInfo, XmCreateDialogMessage, ...

Conseil: dans un programme C, utiliser les fonctions de convénience

Conversion : cas particuliers

XmString

```
   xmString= XmStringCreateSimple(string);
ou xmString= XmStringCreateLtoR(string, XmSTRING_DEFAULT_CHARSET);

   XmStringFree(xmString);

   if (XmStringGetLtoR(xmString, XmSTRING_DEFAULT_CHARSET, &string)) .
```

Pixmap

```
pixmap= XmGetPixmap(screen, bitmapName, foreground, background);

XmDestroyPixmap(screen, pixmap);
```


Traitement des évènements

Actions

Mécanisme

Modifications

WorkProc

TimeOut

Copier/Coller : sélection

Consultation de la sélection

XtGetSelectionValue(widget, selection, askedType, procedure, clientData, time)

selection est un atome: XA_PRIMARY, XA_SECONDARY ou XmInternAtom(display, string, True)
askedType est un atome: XA_STRING, XA_INTEGER, ... (voir Xatom.h)
procedure: recoit en paramètre la valeur et son type (XTCONVERT_FAIL en cas d'erreur), ainsi que clientData
la procédure est appelée lorsque la sélection a été obtenue (asynchrone).
time: est recupéré de l'événement X ayant provoqué l'opération

Affectation de la sélection

XtOwnSelection(widget, selection, time, convertProc, loseProc, doneProc)

convertProc: est déclanché lorsqu'une application exécute XtGetSelectionValue
loseProc: est déclanché lorsqu'une application exécute XtOwnSelection
doneProc: est déclanché pour libérer la mémoire allouée par convertProc *

XtDisownSelection(widget, selection, time)

* XtMalloc(size), XtCalloc(num, size), XtRealloc(ptr, size), XtFree(ptr), XtNew(type), XtNewString(string)

Action

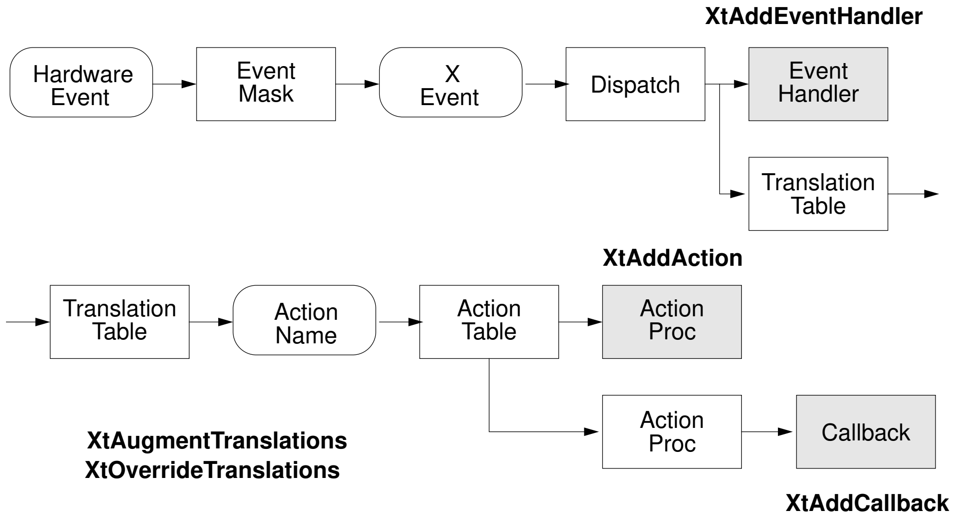
Procédures agissant sur la widget et lancées en réponse à des événements X (en particulier ceux venant d'interactions de l'utilisateur).

internes à la widget = les méthodes (ex : "redraw ()")

associées à des événements X (par table de translations):

- les actions standards (ex : "arm()", "activate()")
- les actions créées par l'utilisateur (globales à toutes les widgets)

Mécanisme



Mise en œuvre des Callbacks

Définition de la *callback*

```
void PrintCallback(widget, clientData, callData);
   Widget widget;
   XtPointer clientData;
   XtPointer callData;
{
   char *text= (char *) clientData;
   fprintf(stderr, "%s\n", text);
}
```

Enregistrement de la *callback*

```
XtAddCallback(helloButton, XmNactivateCallback,
             PrintCallback, "Hello World");

XtAddCallback(quitButton, XmNactivateCallback,
             PrintCallback, "That's all folks");
```


Popup : Mise en œuvre des EventHandlers

Définition de l'*EventHandler*

```
void Postit(widget, clientData, event)
    Widget widget;
    XtPointer clientData;
    XEvent * event;
{
    Widget popup= (Widget) clientData;
    if (event-> xbutton.button == Button2) {
        XmMenuPosition(popup, event);
        XtManageChild(popup);
    }
}
```

Enregistrement de l'*EventHandler*

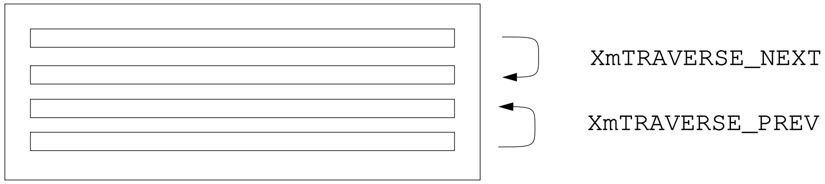
```
XtAddEventHandler(widget,ButtonPressMask,False,Postit,popup);
```

Masque des événements

Pour recevoir les événements non masquables

Traversal


```
XmProcessTraversal(widget, direction)
```




```
ProcessTraversalCB(widget, clientData, callData)
...
{
XmProcessTraversal(widget, clientData);
}

XtAddCallback(text, XmNactivateCallback,
              ProcessTraversal, XmTRAVERSE_NEXT);
```


Time Out

XtAppAddTimeOut

```
void TimerProc (clientData, timerId)
   XtPointer clientData;
   XtIntervalId timerId;
{
   ...
}
timerId= XtAppAddTimeOut(appContext,
                         interval, timerProc, clientData);

XtRemoveTimeOut(timerId);
```

La fonction est appelée après un intervalle indiqué en ms.

Elle n'est appelée qu'une fois.

Modification de la table de translations

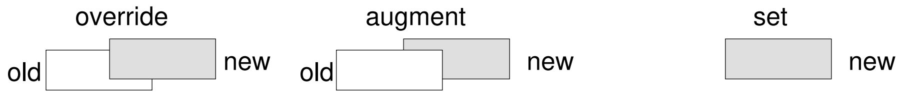
Dans les fichiers de ressources

```
*quit.translations: #override \
<Btn1Down>:    Arm() \n\
<Btn1Up>:      Activate()  Disarm() \n\
```

Dans un programme C

```
char table[]= "\
<Btn1Down>:    Arm() \n\
<Btn1Up>:      Activate()  Disarm() \n\
";
XtTranslations translations= XtParseTranslationTable(table)

   XtOverrideTranslations(widget, translations);
ou XtAugmentTranslations(widget, translations);
ou XtSetValues(widget, /*{XmNTranslation, translations},1 */);
```



Modification de la table d'actions

XtAppAddAction

```
void ActivateAction (widget, event, params, numParams)
   Widget widget;
   XEvent *event;
   String *params;
   Cardinal *numParams;
{
   XmPushButtonCallbackStruct callData;
   callData.reason= XmCR_ACTIVATE;
   callData.event= event;
   XtCallCallbacks (widget, XmNactivateCallback, &callData);
}

XtActionsRec actionTable [ ]= {
   {"Arm", ArmAction},
   {"Disarm", DisarmAction},
   {"Activate", ActivateAction}
}

XtAppAddAction (appContext, actionTable, XtNumber(actionTable));
```

Les actions ainsi ajoutées sont globales à un ApplicationContext.

Les actions prédéfinies sont locales à une classe de Widget.

Classes élémentaires

- Core, Primitive et Manager
- Label, PushButton, ToggleButton
- Frame
- Separator
- Scale
- Text
- List

WorkProc

Une WorkProc est une procédure qui s'exécute en background.

Elle est appelée par XtMainLoop lorsqu'il n'y a pas d'événement X à traiter.

Elle doit être très courte en temps d'exécution.

XtAppAddWorkProc

```
Boolean workProc (clientData)
    XtPointer clientData;
{
    ...
}
```

/* retourne False pour maintenir la procédure */

```
XtWorkProcId workProcId= XtAppAddWorkProc(appContext,
                                          workProc, clientData);
XtRemoveWorkProc(workProcId);
```


Attributs communs à tous les widgets (Core)

Apparence

XmNbackground	Pixel
XmNbackgroundPixmap	Pixmap

Géométrie

XmNx, XmNy	Position
XmNwidth, XmNheight	Dimension

Comportement

XmNtranslations	XtTranslations (selon la classe)
XmNsensitive	Boolean (True)

Classes

- Introduction
  - Historique
  - Introduction
  - Premier exemple
- la boîte à outils "Intrinsics"
  - Documentation
  - Primitives
  - Traitement des événements
- Classes
  - Elémentaires
  - Concernant la Géométrie
  - Menus
  - Elaborées
  - Shell
- Hiérarchie des widgets

PushButton

Sémantique

XmNshownAsDefault

XmNactivateCallback

XmNarmCallback

XmNdisarmCallback

callData

```
void CallbackProc(widget, clientData, callData)
   Widget widget;
   XtPointer clientData;
   XmPushButtonCallbackStruct * callData;
{
   ...
}
```

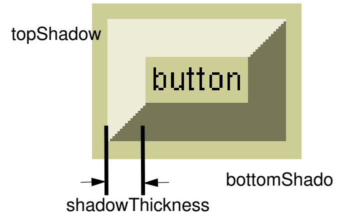
int	reason
XEvent *	event
int	click_count

reason: XmCR_ACTIVATE, XmCR_ARM, XmCR_DISARM

Attributs des widgets MOTIF (Primitive, Manager)

Apparence

XmNtopShadowColor	Pixel
XmNtopShadowPixmap	Pixmap
XmNbottomShadowColor	Pixel
XmNbottomShadowPixmap	Pixmap
XmNshadowThickness	Dimension



Géométrie

XmNunitType {(XmPIXELS), Xm100TH_MILLIMETERS,... }

Comportement

XmNtraversalOn Boolean

pour indiquer que le widget peut recevoir le Keyboard Focus

XmNnavigationType {XmNONE, XmTAB_GROUP,... }

Sémantique

XmNuserData XtPointer (NULL)

ToggleButton

Apparence

XmNvisibleWhenOff Boolean (True)

Comportement

Si créé dans une RadioBox: 1 seul sélectionné à la fois

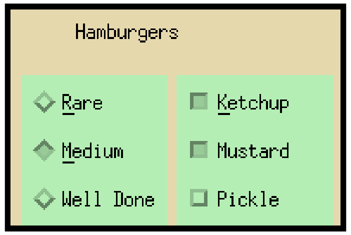
(RadioBox = row column avec attribut “radioBehavior=True”)

Sémantique

XmNset Boolean (False)

XmNarmCallback

XmNdisarmCallback



Label

Apparence

XmNlabelType	{(XmSTRING), XmPIXMAP}
XmNlabelString	XmString
XmNlabelPixmap	Pixmap

Géométrie

XmNmarginLeft, Right, Top, Bottom	Dimension (0, 0, 0, 0)
XmNmarginWidth, Height	Dimension (2, 2)
XmNalignment	{(XmALIGNMENT_CENTER), XmALIGNMENT_BEGINNING, XmALIGNMENT_END}
XmNrecomputeSize	Boolean (True)

Comportement

XmNaccelerator	char * ("")	Effectif uniquement à l'intérieur d'un menu
XmNmenmonic	KeySym()	

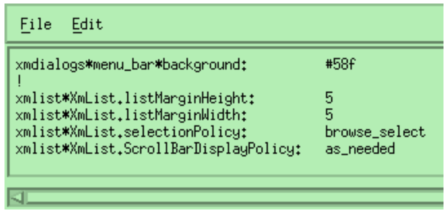
Text

Apparence

- XmNfontList — XmFontList
- XmNcursorPositionVisible — Boolean (True)

Géométrie

- XmNrows, XmNcolumns — short
- XmNresizeWidth — Boolean (False)
- XmNresizeHeight — Boolean (False)



Comportement

- XmNpendingDelete — Boolean (True)
- XmNeditable — Boolean (True)
- XmNeditMode — {(XmSINGLE_LINE_EDIT), XmMULTI_LINE_EDIT}

(préférer TextField)

Sémantique

- XmNvalue — String
- XmNsource — XmTextSource
- XmNactivateCallback
- XmNmodifyVerifyCallback
- XmNmotionVerifyCallback

callData

```
int        reason
XEvent *   event
Boolean    doit
           ...
```

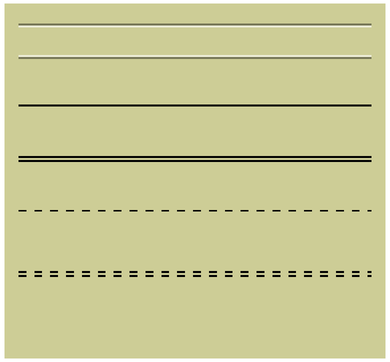
XmTextVerifyCallbackStruc

Separator

Apparence

XmNseparatorType

```
(XmSHADOW_ETCHED_IN)
XmSHADOW_ETCHED_OUT
XmSINGLE_LINE
XmDOUBLE_LINE
XmSINGLE_DASHED_LINE
XmDOUBLE_DASHED_LINE
XmNO_LINE
```



Géométrie

- XmNorientation — {(XmHORIZONTAL), XmVERTICAL}
- XmNmargin — Dimension (0)

Text: functions

XmCreateScrolledText	the Text ScrolledText convenience creation function.
XmCreateText	the Text widget creation function.
XmText	the Text widget class.
XmTextClearSelection	clears the primary selection.
XmTextCopy	copies the primary selection to the clipboard
XmTextCut	copies the primary selection to the clipboard and deletes the selected te
XmTextGetBaseline	accesses the x position of the first baseline
XmTextGetEditable	accesses the edit permission state.
XmTextGetInsertionPosition	accesses the position of the insert cursor
XmTextGetLastPosition	accesses the position of the last text character
XmTextGetMaxLength	accesses the value of the current maximum allowable length of a text string entered from the keyboard.
XmTextGetSelection	retrieves the value of the primary selection.
XmTextGetSelectionPosition	accesses the position of the primary selection
XmTextGetSource	accesses the source of the widget
XmTextGetString	accesses the string value.
XmTextGetTopCharacter	accesses the position of the first character displayed
XmTextInsert	inserts a character string into a text string.
XmTextPaste	inserts the clipboard selection
XmTextPosToXY	accesses the x and y position of a character position
XmTextRemove	deletes the primary selection
XmTextReplace	replaces part of a text string.
XmTextScroll	scrolls text
XmTextSetAddMode	sets the state of Add Mode
XmTextSetEditable	sets the edit permission.
XmTextSetHighlight	highlights text
XmTextSetInsertionPosition	sets the position of the insert cursor
mTextSetMaxLength	sets the value of the current maximum allowable length of a text string entered from the keyboard.
XmTextSetSelection	sets the primary selection of the text.
XmTextSetSource	sets the source of the widget
XmTextSetString	sets the string value.
XmTextSetTopCharacter	sets the position of the first character displayed
XmTextShowPosition	forces text at a given position to be displayed
XmTextXYToPos	accesses the character position nearest an x and y position

Scale

Apparence

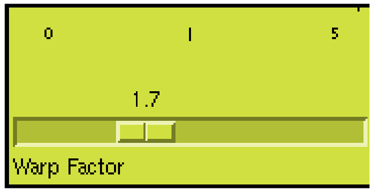
- XmNtitleString — XmString
- XmNshowValue — Boolean (False)
- XmNdecimalPoint — short (0)
- XmNfontList — XmFontList

Géométrie

- XmNorientation — {(XmVERTICAL), XmHORIZONTAL}
- XmNscaleWidth, XmNscaleHeight — Dimension

Comportement

- XmNhighlightOnEnter — Boolean (False)
- XmNhighLightThickness — Dimension (2)



Sémantique

- XmNvalue — int (0)
- XmNminimum, XmNmaximum — int (0, 100)
- XmNdragCallback
- XmNvalueChangedCallback

Frame

Apparence

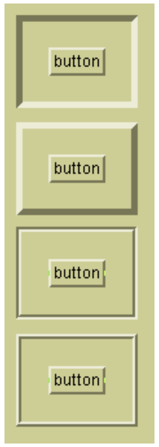
XmNshadowType

XmSHADOW_IN

XmSHADOW_OUT

XmSHADOW_ETCHED_IN

XmSHADOW_ETCHED_OUT



Géométrie

XmNmarginWidth	Dimension (0)
XmNmarginHeight	Dimension (0)

List

Apparence

XmNfontList	XmNfontList

Géométrie

XmNvisibleItemCount	short (1)
XmNlistSpacing	Dimension (0)
XmNlistSizePolicy	{XmCONSTANT, (XmVARIABLE)}

Comportement

XmNselectionPolicy	`{(XmBROWSE_SELECT), XmSINGLE_SELECT, XmMULTIPLE_SELECT, XmEXTENTED_SELECT`

Sémantique

XmNitemCount	int (0)
XmNitems	XmStringTable (NULL)
XmNselectedItemCount	int (0)
XmNselectedItems	XmStringTable (NULL)

XmNbrowseSelectionCallback
XmNsingleSelectionCallback
XmNmultipleSelectionCallbck
XmNextendedSelectionCallback
XmNdefaultActionCallback

List : fonctions

XmList	the List widget class.
XmCreateList	the List widget creation function.
XmCreateScrolledList	the List ScrolledList convenience creation function.
XmListAddItem	adds an item to the list.
XmListAddItemUnselected	adds an item to the list.
XmListAddItems	adds items to the list.
XmListDeleteAllItems	deletes all items from the list.
XmListDeleteItem	deletes an item from the list.
XmListDeleteItems	deletes items from the list.
XmListDeleteItemsPos	deletes items from the list starting at the given position.
XmListDeletePos	deletes an item from a list at a specified position.
XmListDeselectAllItems	unhighlights and removes all items from the selected list.
XmListDeselectItem	deselects the specified item from the selected list.
XmListDeselectPos	deselects an item at a specified position in the list.
XmListGetMatchPos	returns all instances of an item in the list.
XmListGetSelectedPos	returns the position of every selected item in the list.
XmListItemExists	checks if a specified item is in the list.
XmListItemPos	returns the position of an item in the list.
XmListReplaceItems	replaces the specified elements in the list.
XmListReplaceItemsPos	replaces the specified elements in the list.
XmListSelectItem	selects an item in the list.
XmListSelectPos	selects an item at a specified position in the list.
XmListSetAddMode	sets add mode in the list.
XmListSetBottomItem	makes an existing item the last visible item in the list.
XmListSetBottomPos	makes a specified item the last visible item in the list.
XmListSetHorizPos	scrolls to the specified position in the list.
XmListSetItem	makes an existing item the first visible item in the list.
XmListSetPos	makes the item at the given position the first visible position in the lis .

Géométrie

Système de coordonnées

Primitives

Classes spécialisées:

- DrawingArea
- ScrolledWindow et Scrollbar
- RowColumn
- Form
- MainWindow

Système de coordonnées

Axes (0,0) → X

↓ Y

Unités

XmNunitType défini dans les classes XmManager et XmPrimitive

(XmPIXELS)	
Xm100TH_MILLIMETERS	
Xm100TH_INCHES	2.54 cm
Xm100TH_POINTS	point typographique = 1/72 pouce
Xm100TH_FONT_UNITS	selon la font donnée par: *font ou -fn

Primitives

XtSetValues XmNx, XmNy, XmNwidth, XmNheight, XmNborderWidth

XtMoveWidget(widget, x, y)

XtResizeWidget(widget, width, height, borderWidth)

result= XtMakeGeometryRequest(widget, request, reply)

result:
- XtGeometryDone
- XtGeometryYes
- XtGeometryAlmost
- XtGeometryNo

request, reply: `XtGeometryRequest`

```
mode
x, y
width, height
borderWidth
sibling
stackMode
```


DrawingArea

Géométrie:

XmNresizePolicy `{(XmRESIZE_ANY), XmRESIZE_GROW, XmRESIZE_NONE`

Indique comment la taille du DrawingArea peut evoluer.

XmNmarginWidth, XmNmarginHeight `Dimension (10)`

Extension:

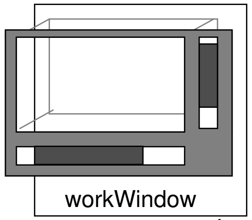
XmNexposeCallback

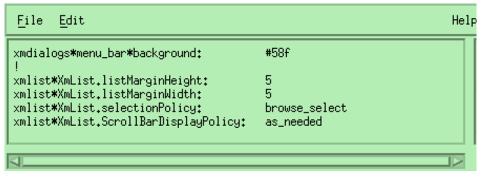
XmNinputCallback

XmNresizeCallback

ScrolledWindow

Principe





Attributs

XmNscrollingPolicy `{XmAUTOMATIC, (XmAPPLICATION_DEFINED`

XmNscrollBarDisplayPolicy `{XmAS_NEEDED, XmSTATIC}`

Hiérarchie

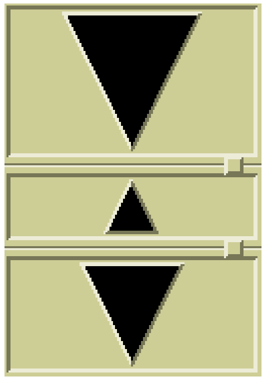
ScrolledWindow → `horizontalScrollBar`, `clipWindow`, `verticalScrollBar`

`clipWindow` → **workWindow**

PanedWindow

Attribut de la PanedWindow

XmNrefigureMode — Boolean (True)



Attributs supplémentaires des fils

XmNallowResize — Boolean (False)
  réponse aux demandes de changement de taille du fils

XmNskipAdjust — Boolean (False)
  indique qu'il ne faut pas retailler le fils

XmNpaneMinimum, XmNpaneMaximum — int (1, 1000)

Scrollbar

Géométrie

XmNorientation — {(XmVERTICAL), XmHORIZONTAL}

Sémantique

XmNminimum, XmNmaximum — int (0, 100)

XmNvalue — int (0)

XmNincrementCallback, XmNdecrementCallback
XmNincrement, XmNdecrement — int (1)

XmNpageIncrementCallback, XmNpageDecrementCallback
XmNpageIncrement, XmNpageDecrement — int (10)

XmNtoTopCallback, XmNtoBottomCallback

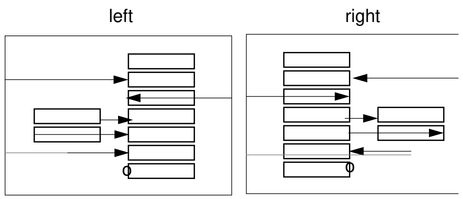
XmNdragCallback, XmNvalueChangedCallback

Form

Attributs supplémentaires des fils

XmNresizable — Boolean (True)

XmNleftAttachment
```
(XmATTACH_NONE)
XmATTACH_FORM
XmATTACH_OPPOSITE_FORM
XmATTACH_WIDGET
XmATTACH_OPPOSITE_WIDGET
XmATTACH_POSITION
XmATTACH_SELF
```



XmNleftPosition — int (0) — (pourcentage)

XmNleftWidget — Widget

XmNleftOffset — int (0)

et de même pour right, top et bottom

RowColumn

Géométrie

XmNorientation — {(XmVERTICAL), XmHORIZONTAL}

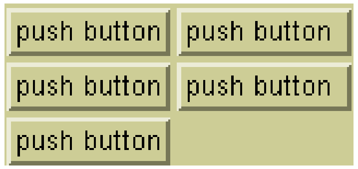
XmNpacking — {XmPACK_TIGHT, XmPACK_COLUMN, XmPACK_NONE}

XmNnumColumns — short (1)

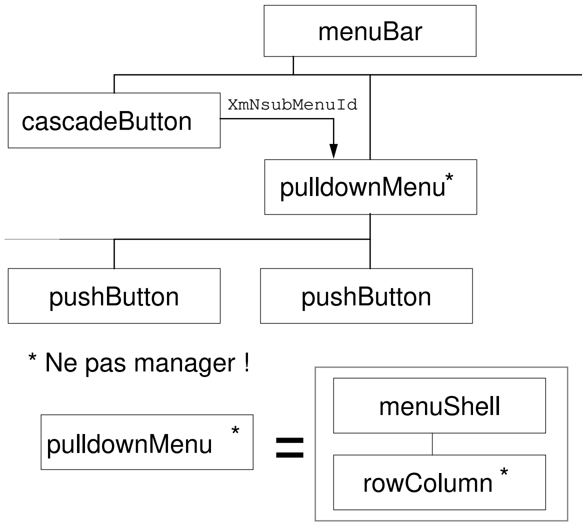
Comportement

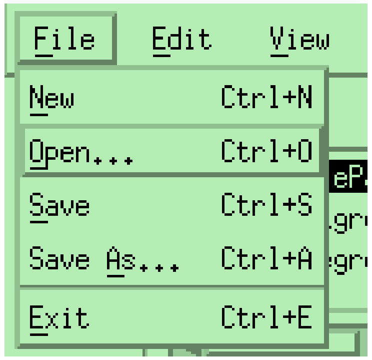
XmNradioBehavior — Boolean (False)

XmNrowColumnType
```
(XmWORK_AREA)
XmMENU_BAR
XmMENU_PULLDOWN
XmMENU_POPUP
XmMENU_OPTION
```



MenuBar, CascadeButton et PulldownMenu (1)

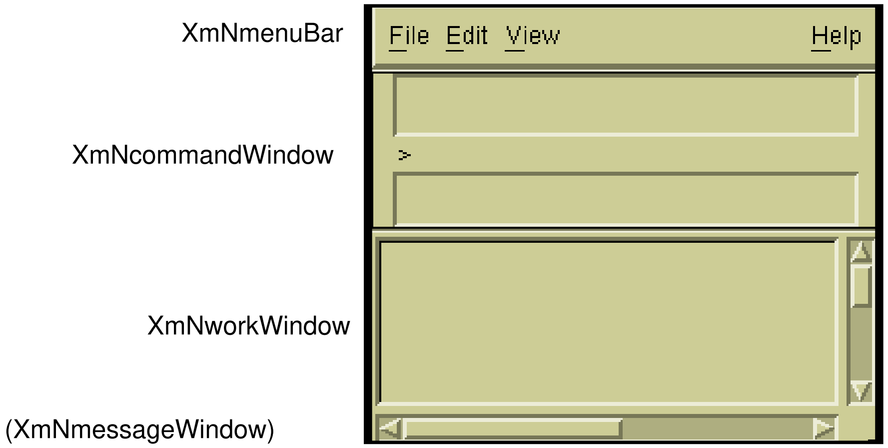




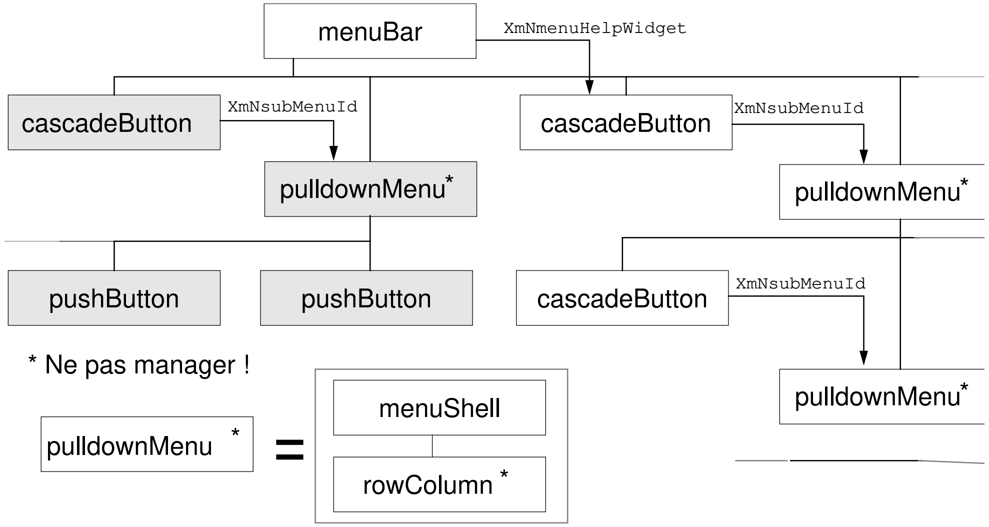
Création : XmCreateMenuBar

MainWindow

sous classe de ScrolledWindow



MenuBar, CascadeButton et PulldownMenu (2)



Menus

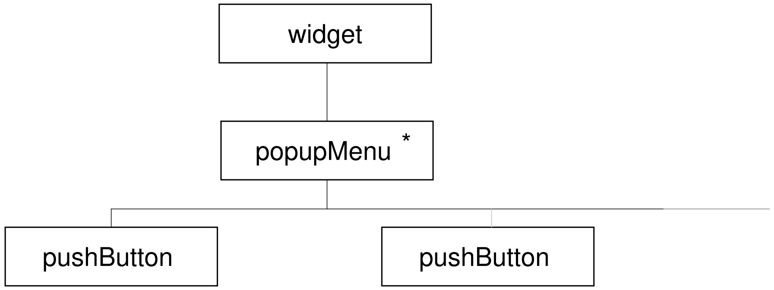
MenuBar, CascadeButton et PulldownMenu

PopupMenu

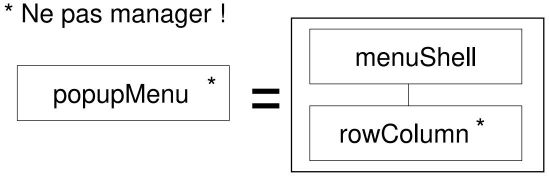
OptionMenu

Accelerator et Mnemonic

PopupMenu : création



* Ne pas manager !



Création : XmCreatePopupMenu

Popup : mise en œuvre des EventHandlers

Définition de l'*EventHandler*

```
void Postit(widget, clientData, event)
     Widget widget;
     XtPointer clientData;
     XEvent * event;
{
     Widget popup= (Widget) clientData;
     if (event-> xbutton.button == Button3) {
          XmMenuPosition(popup, event);
          XtManageChild(popup);
     }
}
```

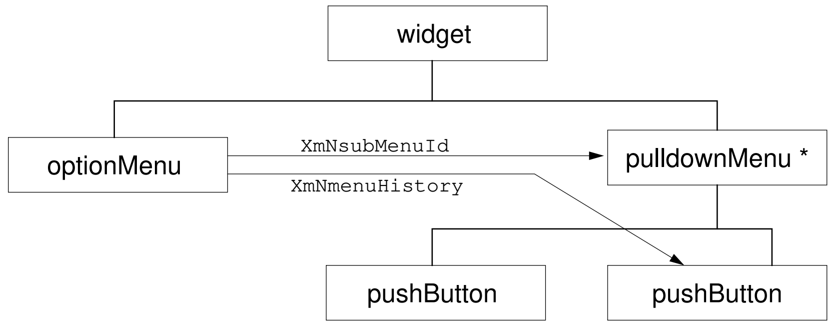
Enregistrement de l' *EventHandler*

```
XtAddEventHandler(widget,ButtonPressMask,False,Postit,popup);
```

Masque des événements

Pour recevoir les événements non masquables

OptionMenu



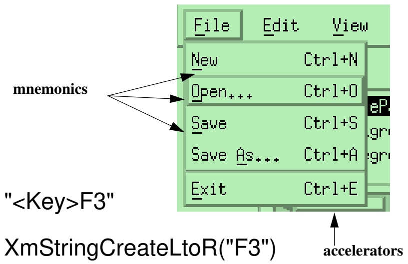
*** Ne pas manager !**

Création : XmCreateOptionMenu

Accelerator et Mnemonic

Attributs de Label, hérités par
PushButton, CascadeButton et ToggleButton

Effectifs seulement à l'intérieur d'un Menu



Accelerator, disponible à tout moment :

XmNaccelerator	**char ***	exemple: "<Key>F3"
XmNacceleratorText	**XmString**	exemple: XmStringCreateLtoR("F3")

Mnemonic, disponible quand le menu est visible:

XmNmnemonic	**KeySym**	exemple: 'O'

NB: Ne pas confondre XmNaccelerator avec XmNaccelerators (Intrinsics)

Classes élaborées

Composite

MesageBox

MessageDialog: Working, Info, Warning, Error, Question

BulletinBoard

SelectionBox

SelectionDialog: Prompt, Command, FileSelection

Attributs communs à tous les composés (Composite)

XmNnumChildren	Cardinal
XmNchildren	WidgetList
XmNinsertPosition	fonction C retournant un numéro d'ordre

MessageBox

Apparence **XmNmessageString**

XmNokLabelString XmString
XmNhelpLabelString XmString
XmNcancelLabelString XmString

XmNdialogType

```
XmDIALOG_ERROR
XmDIALOG_INFORMATION
XmDIALOG_MESSAGE
XmDIALOG_QUESTION
XmDIALOG_WARNING
XmDIALOG_WORKING
```

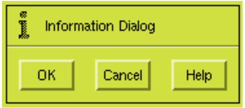
XmNsymbolPixmap Pixmap

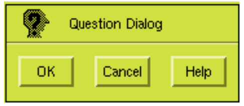
XmNdefaultButtonType

```
(XmDIALOG_OK_BUTTON)
XmDIALOG_CANCEL_BUTTON
XmDIALOG_HELPBUTTON
```

Sémantique **XmNokCallback, cancel, help**

Fonction **XmMessageBoxGetChild**



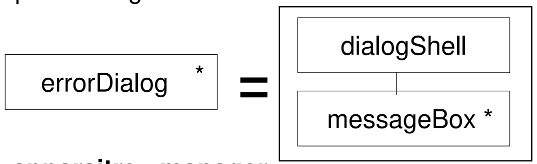


MessageDialog

Création: fonction de convénience

XmCreateErrorDialog
XmCreateInformationDialog
XmCreateMessageDialog
XmCreateQuestionDialog
XmCreateWarningDialog
XmCreateWorkingDialog

* Ne pas manager !



Pour faire apparaitre : manager

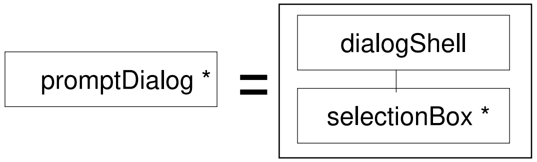
Disparition automatique par ok ou cancel

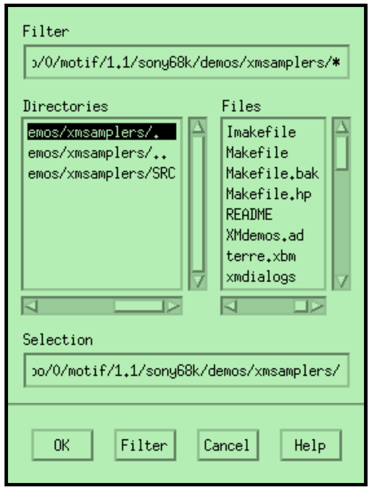
SelectionDialog

Création, fonction de convénience:

XmCreatePromptDialog
XmCreateCommandDialog
XmCreateSelectionDialog
XmCreateFileSelectionDialog

* Ne pas manager !





Pour faire apparaitre : manager

Disparition automatique par ok ou cancel

BulletinBoard

Apparence

XmNshadowType
{(XmSHADOW_OUT), XmSHADOW_IN, XmSHADOW_ETCHED_IN, XmSHADOW_ETCHED_OU

Comportement

XmNautoUnmanage Boolean (false)

XmNdialogStyle

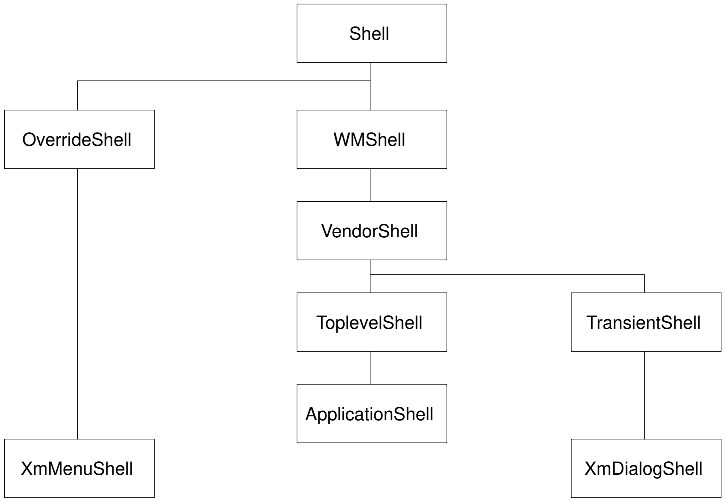
XmDIALOG_SYSTEM_MODAL
XmDIALOG_FULL_APPLICATION_MODAL
XmDIALOG_PRIMARY_APPLICATION_MODAL
XmDIALOG_MODELESS (par défaut si le père est DialogShell)
XmDIALOG_WORK_AREA (par défaut sinon)

Sémantique

XmNmapCallback
XmNunmapCallback

XmNfocusCallback

Shell



SelectionBox

Apparence

XmNlistLabelString	XmString
XmNselectionLabelString	XmString
XmNokLabelString	XmString
XmNhelpLabelString	XmString
XmNcancelLabelString	XmString

Comportement

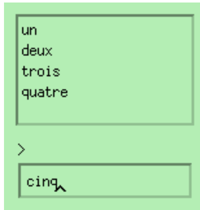
XmNdialogType

XmDIALOG_PROMPT
XmDIALOG_COMMAND
XmDIALOG_SELECTION
XmDIALOG_FILE_SELECTION
XmDIALOG_WORK_AREA

Sémantique

XmNtextString	XmString
XmNlistItemCounts	int (0)
XmNlistItems	XmStringTable
XmNmustMatch	Boolean (False)
XmNnoMatchCallback	

XmNokCallback, XmNapplyCallback, XmNcancelCallback, XmNhelpCallback



Shell

Apparence

XmNvisual	Visual *
XmNoverrideRedirect	Boolean (True pour OverrideShell)
XmNsaveUnder	Boolean (True pour Override et TransientShell)

Géométrie

XmNallowShellResize	Boolean (False)
XmNgeometry	String

Callbacks spécialisées pour les menus

XmNcreatePopupChildProc	fonction pour créer la hiérarchie du popup à la première invocation
XmNpopupCallback	()
XmNpopdownCallback	()

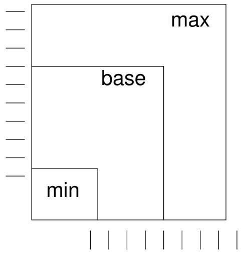
WMShell pour le Window Manager (1)

Apparence

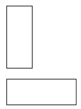
XmNtitle — String (par défaut le nom du widget)

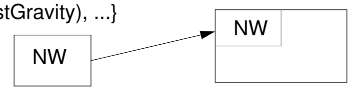
Géométrie

XmNminWidth, XmNminHeight	int
XmNmaxWidth, XmNmaxHeight	int
XmNbaseWidth, XmNbaseHeight	int
XmNwidthIncr, XmNheightIncr	int
XmNminAspectX, XmNminAspectY	int
XmNmaxAspectX, XmNmaxAspectY	int
XmNwinGravity	{ (NorthWestGravity), ...}



ratio X/Y minimum
ratio X/Y maximum





WMShell pour le Window Manager (2)

Comportement

XmNinput	Boolean (True pour TopLevelShell)
XmNtransient	Boolean (True pour TransientShell)
XmNwindowGroup	Widget le leader du groupe de fenêtres
XmNwaitForWM	Boolean (True)
XmNwmTimeOut	int (5000 ms)

Icone

XmNinitialState	{ (NormalState), IconicState}
XmNiconX, XmNiconY	int
XmNiconWindow	Window
XmNiconMask, XmNiconPixmap	Pixmap

VendorShell MOTIF

Apparence

XmNdefaultFontList — XmFontList

XmNmwmDecoration — int (-1)
masque indiquant la présence des éléments de décoration de MWM

Comportement

XmNmwmInputMode — int (-1)
Valeur indiquant: Modeless, PrimaryApplicationModal, FullApplicationModal, SystemMod

XmNmwmFunctions — int (-1)
Masque indiquant la disponiblité des fonctions de MWM sur cette fenêtre

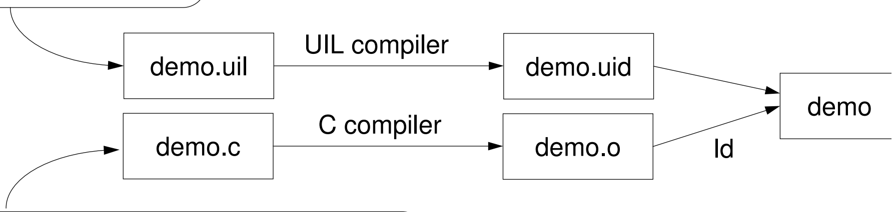
XmNmwmMenu — String
Fonctions ajoutées dans le menu, syntaxe: label [mnemonic] [accelerator] function \n

XmNkeyboardFocusPolicy — (XmEXPLICIT) "click to type"
XmPointer "pointer driven"

UIL

User Interface Language

```
object
 box: XmDrawingArea {
  controls {
   XmPushButton quit;
   XmPushButton hello;
  };
};
```

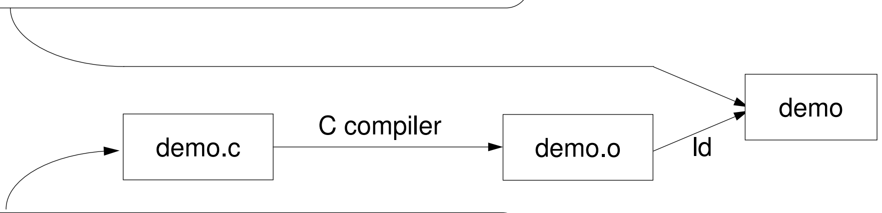



```
toplevel = XtAppCreateShell (...);
MrmInitiatilze ();
MrmOpenHierarchy (...,"demo.uid",...);
MrmFetchWidget (demoHierarchy, "box",
        toplevel, &box, &class);
```


WCL

Widget Creation Library

```
demo.box.wcConstructor: XmCreateDrawingArea
demo.box.wcChildren: quit, hello
demo.box.quit.wcConstructor: XmCreatePushButton
demo.box.quit.x: 50
demo.box.quit.y: 20
```




```
toplevel = XtAppCreateShell (...);
widget = XtNameToWidget(toplevel, "box.quit");
```


Sous classes de VendorShell

TransientShell

XmNtransientFor	Widget

TopLevelShell

XmNiconName	String
XmNiconic	Boolean (False)

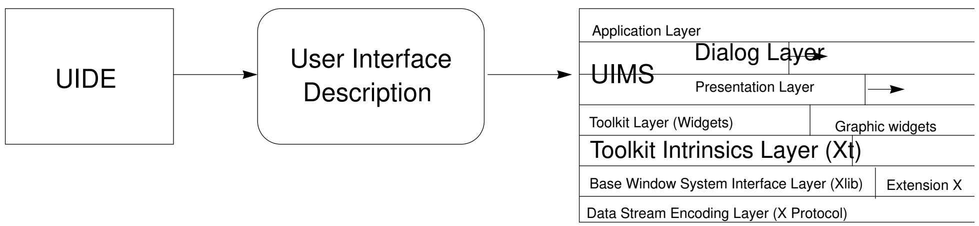
ApplicationShell

XmNargc	int
XmNargv	String *

Au dessus de la toolkit MOTIF

UIMS: User Interface Management System (run-time)

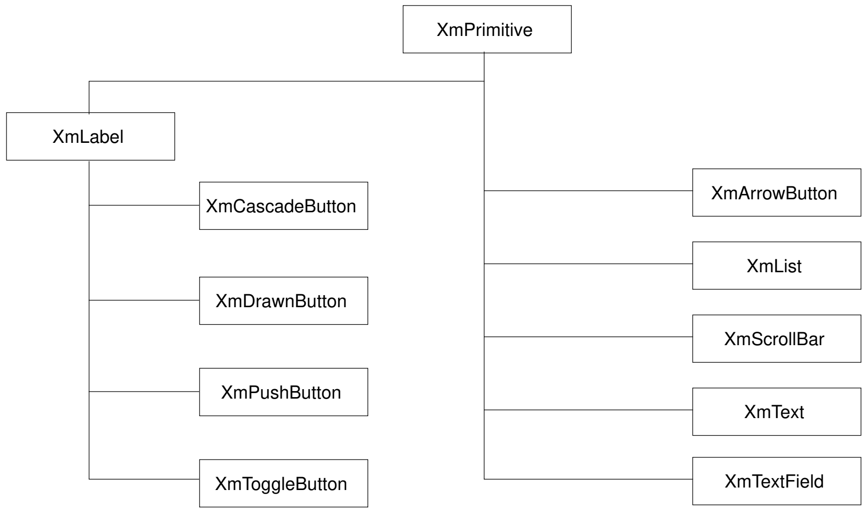
UIDE: User Interface Design Environment (editor)



Etat de l'art : Editeur interactif de la couche présentation (layout editor)

Futur : Outils de description du comportement de l'Interface

Hiérarchie : primitive widget classes



Hiérarchie des widgets

basic widget classes

primitive widget classes

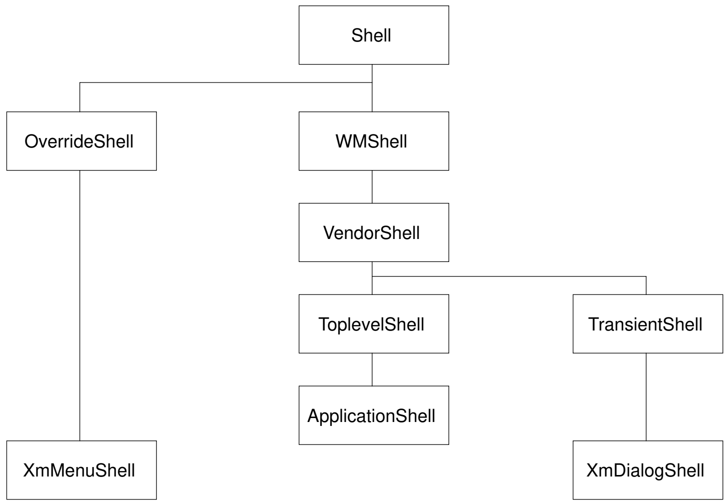
shell widget classes

manager widget classes

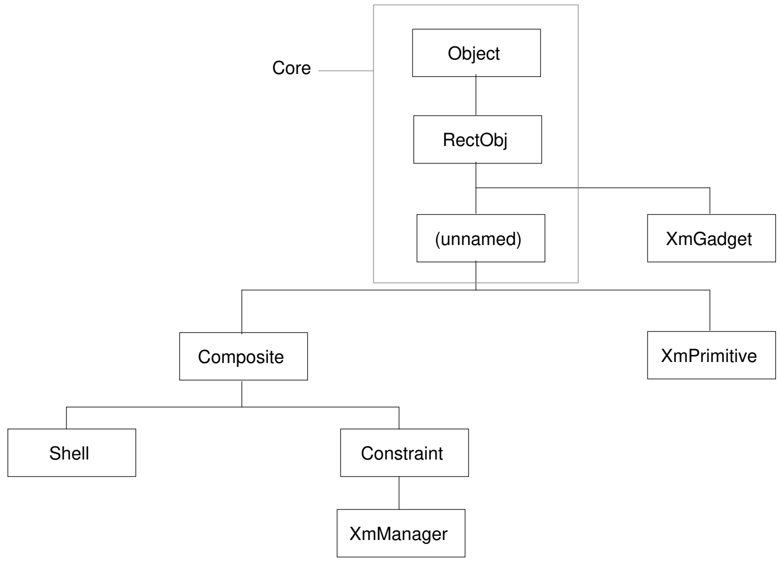
dialog widget classes

gadget classes

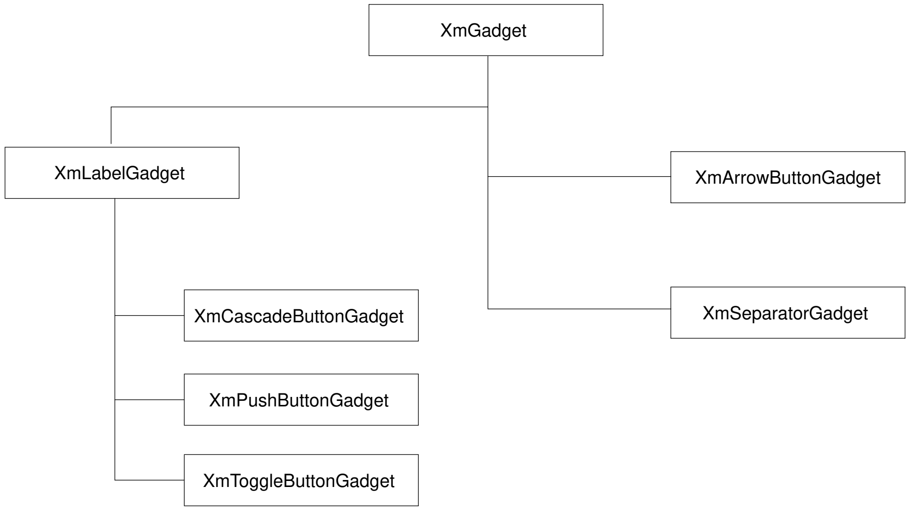
Hiérarchie : shell widget classes



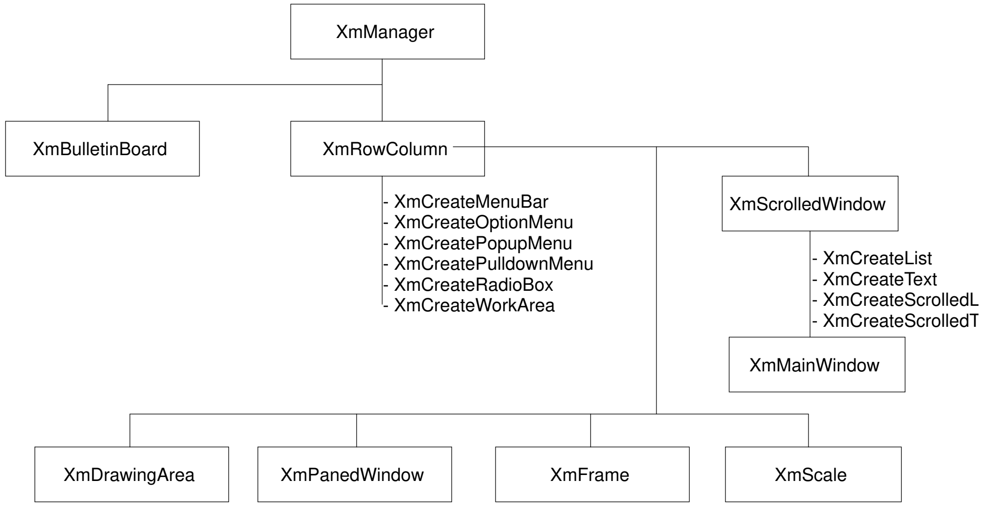
Hiérarchie : basic widget classes



Hiérarchie : gadget classes



Hiérarchie : manager widget classes



Hiérarchie : dialog widget classes