

### 3. Motif : les widgets d'affichage

Widgets primitives

## Motif: les widgets primitives

- o Création de widgets
- o Etiquettes
- o Boutons
- o Textes
- o Variateurs
- o Listes

1

Jean Berstel - Institut Gaspard Monge, Université Marne-la-Vallée

Widgets primitives

## Création de widgets

Dans les fonctions de création Motif, les places du *nom* et de la *mère* sont inversées.

```
XmCreateLabel(mère, nom, args, nargs)
XtCreateWidget(nom, xmLabelWidgetClass, mère, args, nargs)
XtVaCreateWidget(nom, xmLabelWidgetClass, mère, ..., NULL)

XmCreateFileSelectionBox(mère, nom, args, nargs)
```

---

Chaque classe de widgets Motif possède sa propre fonction de création.

- | Elles sont en général à arguments fixes.
- | Elles créent mais ne "managent" pas.
- | Elles sont *nécessaires* quand les widgets sont *composées* de plusieurs widgets simples : FileSelectionBox.
- | Elles sont *utiles* si une widget est un cas particulier d'une autre :
  - les barres de menu sont des RowColumn horizontales
  - les menus sont des RowColumn verticales,...

2

Jean Berstel - Institut Gaspard Monge, Université Marne-la-Vallée

### 3. Motif : les widgets d'affichage

Widgets primitives

## Label

#include <Xm/Label.h>

```
XtVaCreateManagedWidget(nom, XmLabelWidgetClass, mère, ..., NULL)
XmCreateLabel(mère, nom, args, nargs)
```

Etiquette comportant une chaîne composée (de type XmString) ou un pixmap.

---

Core

Primitive

Label

XmNbackground  
XmNborderwidth  
XmNmappedWhenManaged

XmNforeground  
XmNhighlightThickness  
XmNshadowThickness  
XmNhighlightColor

XmNlabelType  
XmNlabelString  
XmNlabelPixmap  
XmNalignment

⊕

Héritage des ressources pour label

Jean Berstel - Institut Gaspard Monge, Université Marne-la-Vallée

Widgets primitives

## Exemples

- o Une étiquette avec un texte fixe :
 

```
XtVaCreateManagedWidget("etiq", XmLabelWidgetClass, mere,
XtVaTypedArg, XmNlabelString, XmRString,
"Taille :", 9, NULL )
```

```
*etiq.labelString : Taille :
```
- o Une étiquette avec pixmap:
 

```
XtVaCreateManagedWidget("etiq", XmLabelWidgetClass, mere,
XmNlabelType, XmPIXMAP,
XtVaTypedArg, XmNlabelPixmap, XmRString, "xi.btm", 7, NULL )
```

```
*etiq.labelType : xmPIXMAP
*etiq.labelPixmap : xi.btm
```

Jean Berstel - Institut Gaspard Monge, Université Marne-la-Vallée

### 3. Motif : les widgets d'affichage

Widgets primitives

## Ressources d'une étiquette

- 4 catégories de ressources, selon leur *origine* ou *destination* :
  - | héritées de `Core`
  - | héritées de `Primitive`
  - | servant directement `Label`
  - | servant aux classes dérivées boutons (`CascadeButton`, `DrawnButton`, `PushButton`, `ToggleButton`).
- Les ressources de `Core` ne sont pas spécifiques à Motif.
- Les ressources de `Primitive` servent à la *décoration* : ombres et leur épaisseur.
- Les ressources de `Label` pour le libellé.

5

Jean Berstel - Institut Gaspard Monge, Université Marne-la-Vallée

Widgets primitives

## Ressources : dimensions

Diagram illustrating the dimensions and resources of a label widget. The diagram shows a rectangular label with the text "saillant" inside. Various dimensions are labeled: height, width, marginTop, marginBottom, marginLeft, marginRight, marginWidth, marginHeight, shadowThickness, and alignment. A callout box lists alignment options: `XmALIGNMENT_BEGINNING`, `XmALIGNMENT_CENTER`, and `XmALIGNMENT_END`. Another callout box explains that `recomputerSize` (true by default) implies that the size is calculated based on the label.

6

Jean Berstel - Institut Gaspard Monge, Université Marne-la-Vallée

### 3. Motif : les widgets d'affichage

Widgets primitives

## Ressources géométriques

XmNx	Position	Core	abscisse par rapport à la mère
XmNy	Position	Core	ordonnée par rapport à la mère
XmNwidth	Dimension	Core	largeur de la fenêtre
XmNheight	Dimension	Core	hauteur de la fenêtre
XmNunitType	enum	Primitive	unité pour valeurs géométriques XmPIXELS Xm100TH_MILLIMETRES Xm100TH_INCHES Xm100TH_POINTS Xm100TH_FONT_UNITS
XmNshadowThickness	short	Primitive	Epaisseur de l'ombre
XmNmarginWidth	short	Label	Largeur de la marge générale
XmNmarginHeight	short	Label	Hauteur de la marge générale
XmNmarginTop	short	Label	Marge additionnelle haute
XmNmarginBottom	short	Label	- basse
XmNmarginLeft	short	Label	- gauche
XmNmarginRight	short	Label	- droite
XmNalignment	short	Label	Alignement du texte/dessin XmALIGNMENT_BEGINNING XmALIGNMENT_CENTER XmALIGNMENT_END

Jean Berstel - Institut Gaspard Monge, Université Marne-la-Vallée

Widgets primitives

## PushButton

```
#include <Xm/PushB.h>
```

```
XtVaCreateManagedWidget(nom, XmPushButtonWidgetClass, mère, ..., NULL)
XmCreatePushButton(mère, nom, args, nargs)
```

Crée un bouton qui réagit au "clic" de la souris.

```

XmNactivateCallback
XmNarmCallback
XmNdisarmCallback
XmNmultiClick
XmNshowAsDefault
                    
```

```

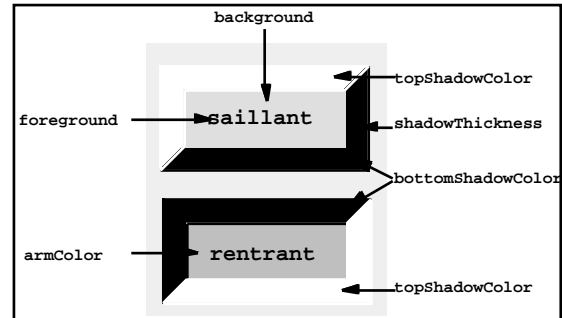
| arm : au moment où on appuie dans la widget.
| disarm : au moment où on relâche.
| activate : au moment où on relâche, dans la
| widget.
|
| multiClick : prend en compte les clics
| multiples.
| showAsDefault : marque ce bouton d'un
| cadre indiquant qu'il est le bouton par défaut.
                    
```

Jean Berstel - Institut Gaspard Monge, Université Marne-la-Vallée

### 3. Motif : les widgets d'affichage

#### Ressources couleurs

- o Les ressources régissant les couleurs sont soit définies dans `Primitives` (pour la décoration en général), soit dans `PushButton` (pour la couleur lorsque le bouton est armé).
- o Le plus souvent, une couleur peut être remplacée par un pixmap : ceci est utile surtout pour des écrans n'ayant que peu de couleurs ou de niveaux de gris.



Jean Berstel - Institut Gaspard Monge, Université Marne-la-Vallée

9

#### Désactiver un bouton / Pixmap

Une fonction `Intrinsic` permet de désactiver (ou réactiver) un bouton:

`XtSetSensitive(w, False)` ou `XtSetSensitive(w, True)`

- o Indispensable pour une interface cohérente (menus par exemple).
- o La ressource `XmNsensitive` (de type booléen, par défaut `True`) est une ressource de `Core`.
- o L'effet par défaut est de griser le texte. Obtenu en appliquant un filigrane ("stipple") au texte.
- o On peut également afficher un pixmap spécifique.

Jean Berstel - Institut Gaspard Monge, Université Marne-la-Vallée

10

### 3. Motif : les widgets d'affichage

Widgets primitives

## Réflexes

- Trois *événements sémantiques* sont définis pour les boutons  
activer, armer, désarmer.
- Trois *listes de fonctions réflexes* dans la classe PushButton, soit :  
XmNactivateCallback  
XmNarmCallback  
XmNdisarmCallback
- Les *fonctions réflexes* sont définies par l'utilisateur, et rangées par lui dans la liste appropriée.

Jean Berstel - Institut Gaspard Monge, Université Marne-la-Vallée

11

Widgets primitives

## ArrowButton

```
#include <Xm/ArrowB.h>
```

```
XtVaCreateManagedWidget(nom, XmArrowButtonWidgetClass, mère, ..., NULL)
XmCreateArrowButton(mère, nom, args, nargs)
```

Crée un bouton avec un triangle indiquant une direction.

---

Ressources :

```
XmNactivateCallback
XmNarmCallback
XmNdisarmCallback
XmNmultiClick
XmNarrowDirection
```

```
XmARROW_UP
XmARROW_DOWN
XmNARROW_LEFT
XmNARROW_RIGHT
```

Jean Berstel - Institut Gaspard Monge, Université Marne-la-Vallée

12

### 3. Motif : les widgets d'affichage

Widgets primitives

## Togglebutton

```
#include <Xm/ToggleB.h>
```

```
XtVaCreateManagedWidget(nom, XmToggleButtonWidgetClass, mère, ..., NULL)
XmCreateToggleButton(mère, nom, args, nargs)
```

Crée un bouton bascule, susceptible de prendre deux états : *sélectionné* ou non.

```

graph TD
    Core[Core] --> Primitive[Primitive]
    Primitive --> Label[Label]
    Label --> ToggleButton[ToggleButton]
            
```

- o Deux comportements de boutons bascule :
  - l Boutons à **cocher** ("check" ou *n-of-many*) : dans un groupe, plusieurs boutons peuvent être sélectionnés simultanément
  - l Boutons **radio** ("radio" ou *one-of-many*) : dans un groupe, un seul bouton est sélectionné à la fois.
  - l Selon le comportement, un *indicateur* différent est affiché.
- o Ce comportement découle du **conteneur** choisi : `XmRadioBox` ou `XmCheckBox`. Des utilitaires pour créer des boutons à bascule par groupes existent. (voir les conteneurs).
- o Chaque bascule comporte, en plus de son indicateur, une étiquette.

Jean Berstel - Institut Gaspard Monge, Université Marne-la-Vallée

13

Widgets primitives

## Ressources

- o Nouvelles ressources (une sélection) :

<code>XmNset</code>	Boolean	<b>bouton sélectionné (False)</b> <b>couleur (éventuelle) d'un bouton sélectionné</b> <b>on prend la couleur si sélectionné</b> <b>radio ou check :</b> <code>XmN_OF_MANY</code> , <code>XmONE_OF_MANY</code> <b>Si vrai, l'indicateur est visible; sinon, seule l'étiquette est vue</b> <b>L'indicateur reste visible si le bouton n'est pas sélectionné</b> <b>Distance à l'étiquette(4)</b>
<code>XmNselectColor</code>	Pixel	
<code>XmNfillOnSelect</code>	Boolean	
<code>XmNindicatorType</code>	unsigned char	
<code>XmNindicatorOn</code>	Boolean	
<code>XmNvisibleWhenOff</code>	Boolean	
<code>XmNspacing</code>	Dimension	

- o Fonctions utilitaires :

```
Boolean XmToggleButtonGetState(Widget w) : retourne l'état
XmToggleButtonSetState(Widget w, Boolean s, Boolean notifier) :
```

si `notifier` est vrai, les réflexes enregistrées sur `XmvalueChangedCallback` sont exécutées

Jean Berstel - Institut Gaspard Monge, Université Marne-la-Vallée

14

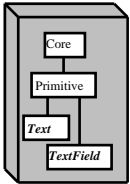
### 3. Motif : les widgets d'affichage

Widgets primitives

## Text - TextField

```
#include <Xm/Text.h>
#include <Xm/TextF.h>
```

```
XmCreateText(mère, nom, args, nargs)
XmCreateTextField(mère, nom, args, nargs)
XmCreateScrolledText(mère, nom, args, nargs)
```



```
graph TD
  Core --> Primitive
  Primitive --> Text
  Text --> TextField
```

- o Créé une widget texte :
  - | une ligne de texte (TextField)
  - | une fenêtre avec (Scrolled) ou sans ascenseurs, et les possibilités d'un éditeur de texte.
- o Pour une widget de la classe Text ou ScrolledText :
  - | XmNeditMode est
    - XmSINGLE\_LINE\_EDIT pour un texte à une ligne (défaut), même pour un Text (bogue ?).
    - XmMULTI\_LINE\_EDIT pour un nombre variable de lignes.

Jean Berstel - Institut Gaspard Monge, Université Marne-la-Vallée

15

Widgets primitives

## Textes

- o Exemple :

```
texte = XmCreateScrolledText(tete, "texte", NULL, 0);
XtVaSetValues(texte,
  XmNrows,      10,
  XmNcolumns,   80,
  XmNeditMode,  XmMULTI_LINE_EDIT,
  XmNautoShowCursorPosition, TRUE,
  NULL);
```

Déroule la fenêtre si nécessaire pour toujours montrer le curseur d'insertion

Jean Berstel - Institut Gaspard Monge, Université Marne-la-Vallée

16

### 3. Motif : les widgets d'affichage

Widgets primitives

- o Pour *recupérer* le contenu de la widget de texte `texte` dans une chaîne de caractères (`char*`), faire :
 

```
chaîne = XmTextGetString(texte);
chaîne = XmTextFieldGetString(texte);
```
- o Exemple (programme `bulletin.c`):

```
texte = XtVaCreateManagedWidget("texte", xmTextWidgetClass,
    mere, NULL);
bouton = XtVaCreateManagedWidget("bouton", xmPushButtonWidgetClass,
    mere, NULL);
XtAddCallback(bouton, XmNactivateCallback, lire, texte);

void lire(Widget w, XtPointer texte, XtPointer cd)
{
    printf("Texte lu: %s\n", XmTextGetString((Widget)texte));
}
```

Jean Berstel - Institut Gaspard Monge, Université Marne-la-Vallée

17

Widgets primitives

## Ressources textes

- o Ressources géométriques

<code>XmNcolumns</code>	short	largeur mesurée en nombre de caractères. (20)
<code>XmNrows</code>	short	hauteur, mesurée en lignes(1)
<code>XmNmaxlength</code>	int	longueur maximale du texte
<code>XmNresizeWidth</code>	Boolean	retailer pour tout montrer (False)
<code>XmNresizeHeight</code>	Boolean	idem, si Multi-line,(False)
<code>XmNfontList</code>	XmFontList	police à utiliser
<code>XmNmarginWidth</code>	short	largeur du bord autour du texte
<code>XmNmarginHeight</code>	short	hauteur du bord autour du texte

- o Ressources de comportement

<code>XmNcursorPosition</code>	XmTextPosition	Position du curseur d'insertion (0)
<code>XmNautoShowCursorPosition</code>	Boolean	scroller le curseur dans la vue
<code>XmNblinkRate</code>	int	temps de clignotement (500 ms)
<code>XmNcursorPositionVisible</code>	Boolean	si le curseur est visible
<code>XmNtopCharacter</code>	int	début d'affichage (0)
<code>XmNeditable</code>	XmFontList	police à utiliser
<code>XmNwordWrap</code>	Boolean	saut de ligne automatique

Jean Berstel - Institut Gaspard Monge, Université Marne-la-Vallée

18

### 3. Motif : les widgets d'affichage

Widgets primitives

## Fonctions et réflexes

- o Pour *écrire* une chaîne de caractères, faire

```
XmTextSetString(w, chaîne);
XmTextFieldSetString(w, chaîne);
```

  - | Les utilitaires Set et Get concernent en fait l'attribut XmNvalue.
- o Listes de réflexes :

XmNactivateCallback	(activée par un retour-chariot)
XmNmodifyVerifyCallback	(juste <b>avant</b> suppression ou insertion)
XmNvalueChangedCallback	(juste <b>après</b> suppression ou insertion)
XmNmotionVerifyCallback	(juste avant un déplacement du curseur d'insertion)
XmNfocusCallback	(juste avant que la widget ne devienne foyer d'entrée)
XmNlosingFocusCallback	(lorsque la widget perd le foyer d'entrée)
XmNgainPrimaryCallback	(lorsque la widget devient propriétaire d'une sélection)
- o Ainsi :

modifyVerifyCallback	peut servir à <i>refuser</i> l'insertion,
activateCallback	peut remplir le rôle d'un bouton d'accord.

Jean Berstel - Institut Gaspard Monge, Université Marne-la-Vallée

19

Widgets primitives

## String et XmString

- o String est synonyme de char\*
- o XmString est une chaîne avec formatage interne.
- o Conversion de String en XmString :

```
XmString xmstring;
...
xmstring = XmStringCreateSimple(chaîne);
...
XmStringFree(xmstring);
```
- o Conversion de XmString en String :

```
char *chaîne;
...
XmStringGetLtoR(xmstring, XmFONTLIST_DEFAULT_TAG, &chaîne);
...
XtFree(chaîne);
```
- o XmFONTLIST\_DEFAULT\_TAG : le "jeu de caractères" par défaut

Jean Berstel - Institut Gaspard Monge, Université Marne-la-Vallée

20

### 3. Motif : les widgets d'affichage

Widgets primitives

## Exemple

```
char *mois[] = {"Janvier", "F\351vrier", "Mars", "Avril", "Mai", "Juin",
"Juillet", "Ao\373t", "Septembre", "Octobre", "Novembre", "D\351cembre"};

int n = XtNumber(mois);
XmStringTable xm_mois; // = XmString *

xm_mois = (XmStringTable)XtMalloc(n * sizeof (XmString *));

for (i = 0; i < n; i++)
    xm_mois[i] = XmStringCreateSimple(mois[i]);
...

for (i = 0; i < n; i++)
    XmStringFree(xm_mois[i]);

XtFree((XtPointer)xm_mois);
```

21

Jean Berstel - Institut Gaspard Monge, Université Marne-la-Vallée

Widgets primitives

## Scale

```
#include <Xm/Scale.h>

XtVaCreateManagedWidget(nom, xmScaleWidgetClass, mère, ..., NULL)
```

Un *variableur* affiche une valeur numérique (entier) compris entre un minimum et un maximum. horizontalement. (En fait un conteneur)

```
graph TD
    Core[Core] --> Composite[Composite]
    Composite --> Constraint[Constraint]
    Constraint --> Manager[Manager]
    Manager --> Scale[Scale]
```

- o La valeur change de façon interactive en agissant sur le *cursor*.
- o Des valeurs non entières peuvent être simulées par un point décimal.
- o L'affichage peut se faire verticalement (défaut) ou horizontalement.

22

Jean Berstel - Institut Gaspard Monge, Université Marne-la-Vallée

### 3. Motif : les widgets d'affichage

Widgets primitives

## Exemple

```
echelle = XtVaCreateManagedWidget("Mois",
    xmScaleWidgetClass, boite,
    XtVaTypedArg, XmNtitleString, XmRString,
        "les mois", 1+strlen("les mois"),
    XmNmaximum, 12,
    XmNminimum, 1,
    XmNvalue, 1, /* valeur initiale */
    XmNshowValue, True, /* afficher valeur */
    XmScaleMultiple, 3, /* increment si on clique hors curseur */
    XmNorientation, XmHorizontal,
    NULL);
```

- o Autre ressource utile :  
XmNprocessingDirection: XmMAX\_ON\_TOP  
XmMAX\_ON\_BOTTOM  
XmMAX\_ON\_LEFT  
XmMAX\_ON\_RIGHT
- o Utilitaires :  
XmScaleGetValue(w, &valeur);  
XmScaleSetValue(w, valeur);

Jean Berstel - Institut Gaspard Monge, Université Marne-la-Vallée

23

Widgets primitives

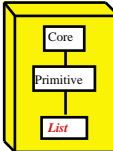
## Listes

```
#include <Xm/List.h>
```

```
XmCreateList(mère, nom, args, nargs)
XmCreateScrolledList(mère, nom, args, nargs)
```

Crée une liste de chaînes de "noms", dans laquelle on peut sélectionner des entrées selon plusieurs modes.

---



```
graph TD
    Core[Core] --> Primitive[Primitive]
    Primitive --> List[List]
```

- o Exemple :  

```
liste = XmCreateScrolledList(racine, "mois", NULL, 0);
XtVaSetValues(liste,
    XmNvisibleItemCount, 5, ← Nombre d'entrées visibles
    XmNitemCount, n, ← Nombre total d'entrées
    XmNitems, xm_mois, ← Texte des entrées (liste de XmString !)
    NULL);
```
- o Les entrées doivent être données en XmString.
- o XmScrolledList est une *widget composée*, formée d'une liste et d'ascenseurs.

Jean Berstel - Institut Gaspard Monge, Université Marne-la-Vallée

24

### 3. Motif : les widgets d'affichage

Widgets primitives

## Listes : stratégies de sélection

Il y a 4 stratégies de sélection (qui se choisissent par `XmNselectionPolicy`) :

**XmSINGLE\_SELECT**

- | la sélection d'une entrée change son état, et désélectionne une autre;
- | *au plus une* entrée sélectionnée à la fois;
- | la sélection se fait quand on *appuie* sur le bouton;
- | sur clavier, sélection par barre d'espace. ("add mode")

**XmBROWSE\_SELECT** (défaut)

- | en sélectionnant une nouvelle entrée, l'ancienne est désélectionnée;
- | il y a *toujours une entrée sélectionnée*.
- | la sélection se fait quand on *relâche* le bouton;
- | sur clavier, sélection pour *chaque* entrée visitée. ("normal mode")

**XmMULTIPLE\_SELECT**

- | la sélection d'une entrée change son état, les autres ne bougent pas;
- | sélection comme pour `SINGLE`.

**XmEXTENDED\_SELECT**

- | on peut choisir des rangées contiguës (par `Ma jBtn1`) ou non (par `CtlBtn1`);
- | sélection quand on relâche le bouton.

Jean Berstel - Institut Gaspard Monge, Université Marne-la-Vallée

25

Widgets primitives

## Utilitaires

- o Les entrées d'une liste sont numérotées à partir de 1.
- o Fonctions utilitaires (il y en a beaucoup)
  - | pour *sélectionner* et *désélectionner* des entrées par programme;
  - | pour *gérer* une liste : ajouter, supprimer, chercher;
  - | pour *recupérer* des informations.
- o Exemple:

```
int *positions;
int n;
Boolean XmListGetSelectedPos(liste, &positions, &n);
```

  - | retourne `True` s'il y a des entrées sélectionnées. Dans ce cas,
  - | alloue un tableau d'entiers `positions` contenant les numéros des entrées sélectionnées;
  - | `n` est le nombre d'entrées sélectionnées.

Jean Berstel - Institut Gaspard Monge, Université Marne-la-Vallée

26

### 3. Motif : les widgets d'affichage

Widgets primitives

- o Pour récupérer les entrées sélectionnées par leur nom, autrement que par une fonction utilitaire, utiliser la ressource `XmNselectedItems`
- o Pour connaître le numéro d'une entrée, prendre `XmListItemPos`

```
void entree(Widget w, XtPointer client_data, XtPointer cd) {
    XmStringTable ch; // = XmString *
    char *choix;
    int i,n;

    XtVaGetValues(w,
        XmNselectedItems, &ch,
        XmNselectedItemCount, &n, NULL);

    printf("%d entrees selectionees :\n", n);
    for (i = 0; i < n; i++) {
        XmStringGetLtoR(ch[i], XmFONTLIST_DEFAULT_TAG, &choix);
        printf("%d (%s)\n", XmListItemPos(w,ch[i]), choix);
        XtFree(choix);
    }
}
```

Jean Berstel - Institut Gaspard Monge, Université Marne-la-Vallée

27

Widgets primitives

### Structures réflexes : généralités

- o A chaque évènement sémantique, Motif associe une *structure réflexe* ("callback structure")
- o A chaque appel d'une fonction réflexe, Motif crée une structure réflexe appropriée qui contient le *pourquoi* et le *comment* de l'évènement sémantique.
- o Motif passe un pointeur vers cette structure réflexe en troisième argument à la fonction réflexe, lors de son appel.
- o La structure réflexe la plus pauvre est `XmAnyCallbackStruct`
- o Le champ `reason` indique la "raison" de l'appel.
- o C'est un nom pour l'évènement sémantique.
- o Motif définit 45 raisons.

```
typedef struct
{
    int    reason;
    XEvent *event;
} XmAnyCallbackStruct;
```

Jean Berstel - Institut Gaspard Monge, Université Marne-la-Vallée

28

### 3. Motif : les widgets d'affichage

Widgets primitives

- o Pour un *bouton*, la structure réflexe est `XmPushButtonCallbackStruct`.
- o Trois "raisons" existent
- o Tout nom symbolique de raison commence par `XmCR`.
  
- o `click_count` contient le nombre de clics dans la dernière séquence de clics multiples . Valide seulement si la raison est `XmCR_ACTIVATE`.

```
typedef struct  
{  
    int     reason;  
    XEvent  *event;  
    int     click_count;  
} XmPushButtonCallbackStruct;
```

```
XmCR_ACTIVATE  
XmCR_ARM  
XmCR_DISARM
```

29

Jean Berstel - Institut Gaspard Monge, Université Marne-la-Vallée