VMS DECwindows Motif V1.0

Overview

Bernard OURGHANLIAN

Symposium DECUS

6 Juin 1991

VMS 20

Sommaire

*** OSF/Motif est une masque déposée de Open Software Foundation**

Introduction

Cette presentation présuppose que l'on a une idée au moins sommaire de DECwindows et de son architecture

Introduction

- Les principales nouvelles fonctionnalités de DECwindows Motif V1.0 sont :
  - -- L’intégration de **OSF/Motif V1.1.1**
  - -- Le *binding* de Motif pour les langages C, Ada, FORTRAN et Pascal
  - -- La compatibilité avec X11 R4
  - -- L’implantation du *backingstore* au niveau du serveur
  - -- L’introduction de **LinkWork**
  - -- Nouvelle extention du serveur avec l’apparition de XIE **(X I**mage **E**xtension)
  - -- L’introduction d’aide avec l’apparition du *DECwindows Motif Help System*
  - -- L’Introduction de CBI (C*omputer Based Instruction*)
  - -- L’amélioration de l’interopérabilité

Introduction

- Le but principal de VMS DECwindows Motif V1.0 est d’incorporer l’inferface Motif de l’OSF
- OSF/Motif ne remplace pas DECwindows mais intègre l’interface utilisateur Motif :

  -- Motif Toolkit Version 1.1.1

  -- Motif Window Manager

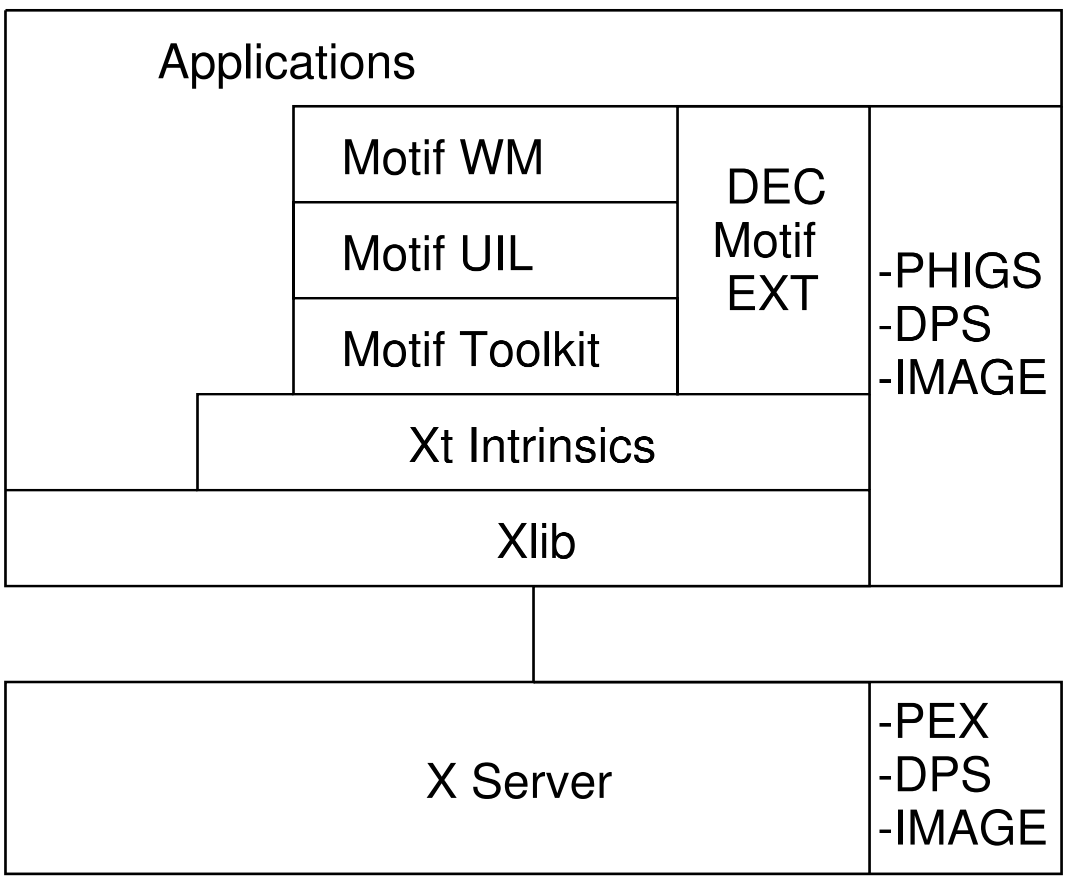
  -- Motif UIL compiler
- Un certain nombre d’outils de portage sont intégrés à DECwindows Motif V1.0 pour aider les développeurs à porter leurs applications utilisant XUI vers Motif

Nouvelles fonctionnalités

Intégration avec OSF/Motif

- L'architecture de DECwindows Motif
- Les composants du Toolkit DECwindows
- Les extensions de DIGITAL à Motif
- Les images partageables du Toolkit
- Les applications portées vers Motif
- Le *Window Manager* Motif

Nouvelles fonctionnalités
L’architecture de DECwindows Motif

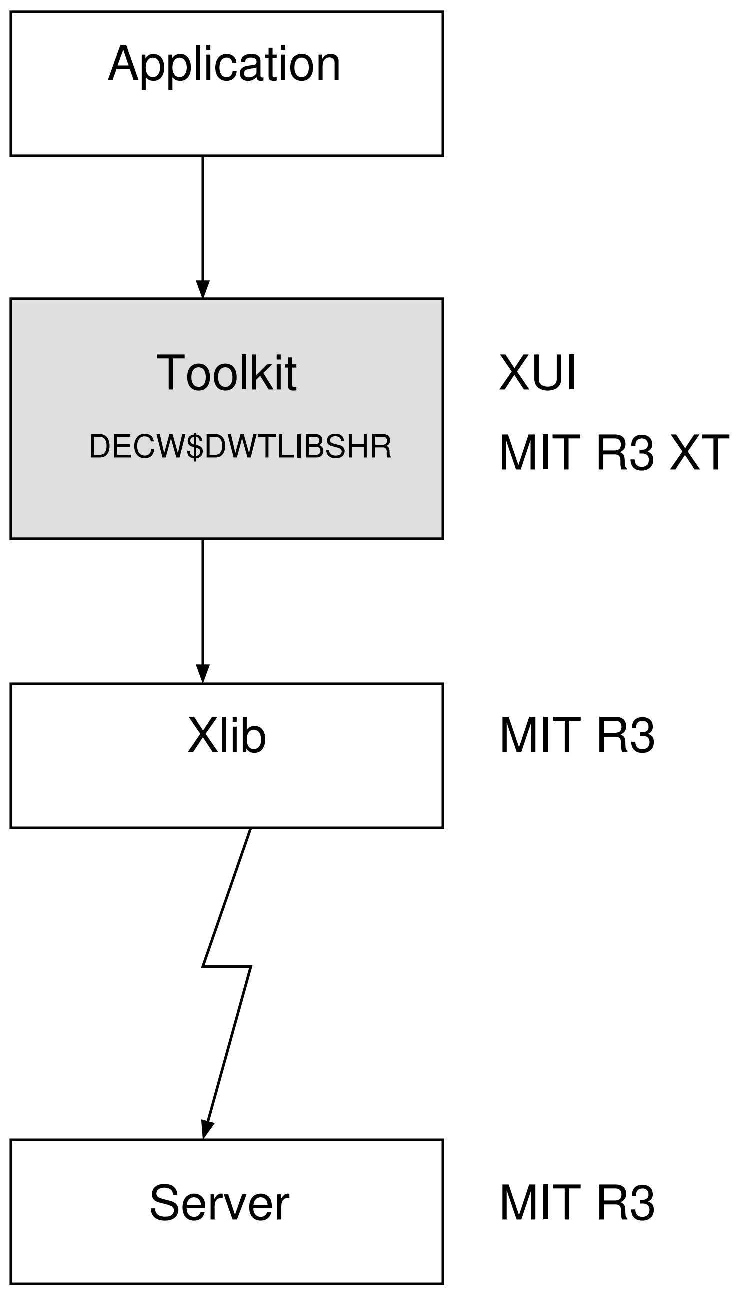


Nouvelles fonctionnalités

Les composants du Toolkit DECwindows

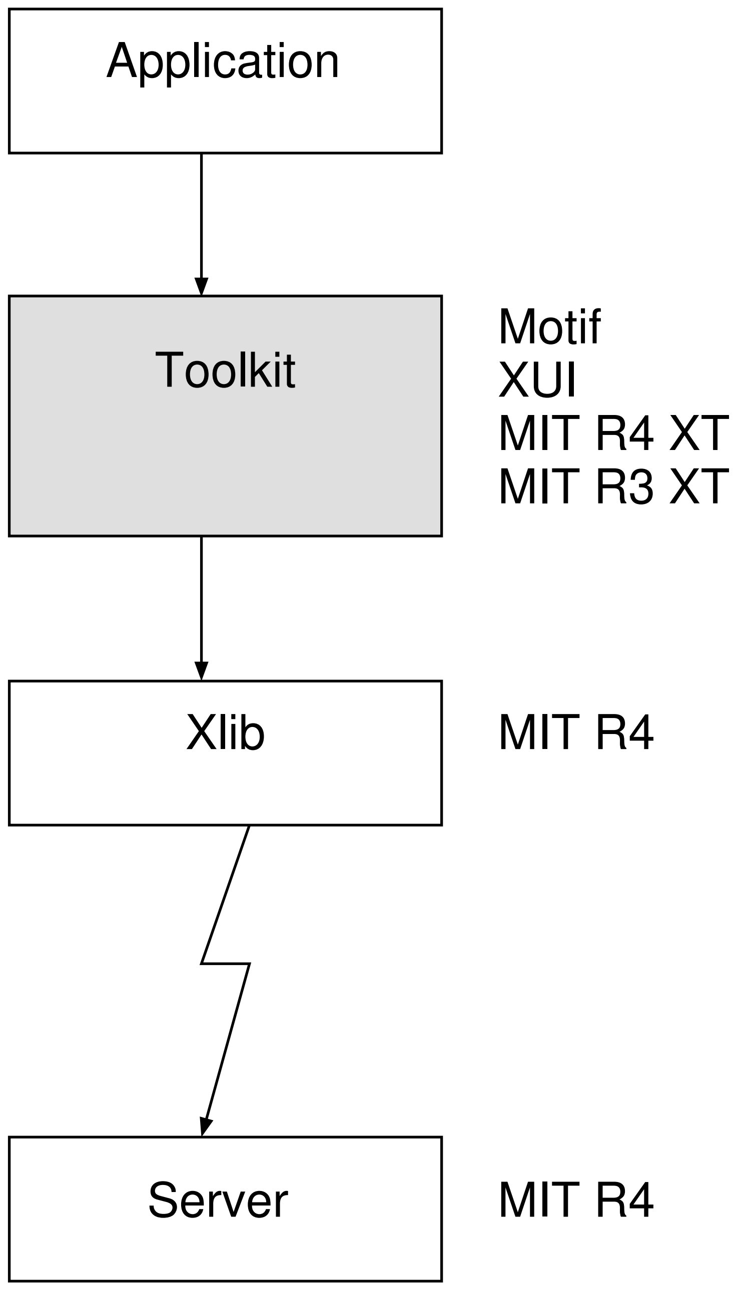
- Le *Toolkit* DECwindows comprend :
  -- Le *Toolkit* Motif (MIT R4 *instrinsics*)
  -- Le compilateur UIL Motif
  -- Le *Ressource Manager* Motif
  -- Les extensions de DIGITAL à Motif (DXm)
  -- Le *Toolkit* XUI (MIT R3 et R4 *intrinsincs*)
  -- Le compilateur UIL XUI
  -- Le *DIGITAL Ressource Manager* (DRM pour XUI)

Nouvelles fonctionnalités
DECToolkit V2



Nouvelles fonctionnalités

Nouveau Toolkit



Nouvelles fonctionnalités

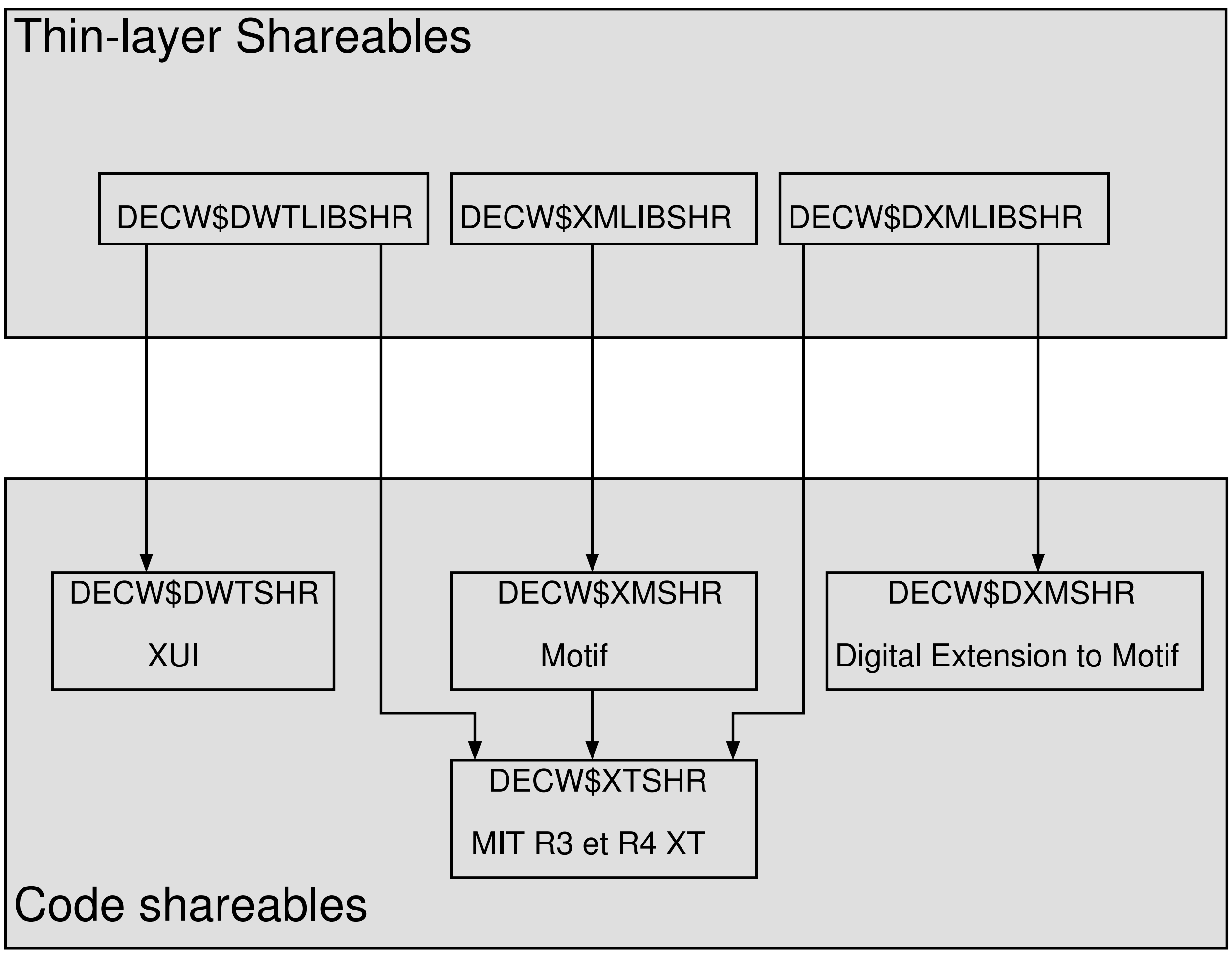
Nouvelle architecture des images partageables

- L'architecture des images partageables du *Toolkit* DECwindows est complètement nouvelle pour :
  - -- Permettre le support d'OSF/Motif
  - -- Permettre la possibilité de mixer l'utilisation des *widgets* Motif et XUI dans la même application (avec quelques restrictions mentionnées plus loin)
  - -- Permettre une **compatibilité ascendante binaire** de la version R3 des *intrinsincs* du MIT et de la version R4 et permettre de jetter les bases du support d'une meilleure compatibilité binaire dans le futur

- 2 classes d'images partageables pour le *Toolkit* DECwindows :
  - -- *thin-layer shareable images* : ce sont les images partageables avec lesquelles se font les éditions de lien
  - -- *code shareable images* : ce sont les images partageables contenant le code du *Toolkit*

Nouvelles fonctionnalités

Nouvelles images partageables du DECtoolkit



Nouvelles fonctionnalités

Nouvelles images partageables du DECtoolkit

SHAREABLE IMAGE	UTILISÉE PAR
DECW$DWTLIBSHR	Applications qui utilisent XUI et Xt
DECW$XMLIBSHR	Applications qui utilisent Xm (OSF/Motif standard)
DECW$DXMLIBSHR	Applications qui utilisent DXm (Digital's Extension to Motif)

Nouvelles fonctionnalités

Nouvelles images partageables du DECtoolkit

- L'image partageable DECW$DWTLIBSHR a été changée pour ne contenir que :
  - -- les symboles "publics" utilisés par les applications utilisant le *Toolkit* :
    - > vecteurs de transferts
    - > données globales
  - -- le code qui active dynamiquement les images partageables concernées
- Il faut donc utiliser :
  - -- DECW$XMLIBSHR pour les applications utilisant Xm (OSF/Motif standard)

-- DECW$DXMLIBSHR pour les applications utilisant DXm (*DIGITAL's extentions* to Motif)

-- DECW$DWTLIBSHR pour les applications utilisant XUI et/ou les *Intrinsics* (Xt)

Nouvelles fonctionnalités

Les images partageables du DECtoolkit

Compatibilité ascendante :

La combinaison de DECW$DWTLIBSHR, DECW$DWTSHR et DECW$XTSHR permet d'assurer une **compatibilité binaire ascendante** avec les versions précédentes avec une **exception** : les applications qui activent dynamiquement DECW$DWTLIBSHR avec LIB$FIND_IMAGE_SYMBOL

- Le problème posé par l'activation dynamique des images est que :
  - -- Dans les versions précédentes de DECwindows, les applications XUI étaient appelées à travers des vecteurs de transferts qui utilisaient la sémantique R3 des *intrinsics*
  - -- Cette sémantique a changé avec R4 de façon **incompatible**

Pour obtenir les *intrinsics* de X11 R3, il ne faut pas refaire d'édition de lien des application

- Les applications qui activaient dynamiquement le *toolkit* doivent arrêter de le faire et doivent faire une édition de lien avec DECW$DWTLIBSHR

Les *"Thin-layer" shareables images sont* petites et activent automatiquement le code du *toolkit* : il n'y a donc pas de raison d'activer dynamiquement DECW$DWTLIBSHR

DIGITAL ne garantit pas une compatibilité ascendante pour les applications qui activent dynamiquement les images partageables du *toolkit*

- Il y a un certain nombre de changements incompatibles entre les *Intrinsics* R3 et R4; en voici les principaux :
  - -- En R3, le nom de l'*application shell* provient du nom du *shell;* en R4, ce paramètre est ignoré
  - -- R3 XtDisplayInitialize accepte un nom d'application "vide"; ce n'est pas le cas en R4
  - -- Un paramètre a été supprimé des appels à XtGetSelectionValueIncremental et XtGetSelectionValuesIncremental
  - -- Un certain nombre de modifications au niveau de la sémantique des *widgets* (modification du passage des arguments, paramètres additionnels)
- Il est souhaitable de consulter la documentation des *intrinsics* en cas de problèmes

Nouvelles fonctionnalités

Les images partageables du DECtoolkit

Restrictions lors de l'utilisation mixte des *widgets* Motif et XUI :

Il y a une restriction sur l'utilisation mixte des *widgets* Motif et XUI :

- Ceci est dû au fait que XUI et Motif utilisent chacun leur propre *Vendor Shell Widget Class* :
  - -- Si une application fait son édition de lien avec XUI mais pas Motif, on utilise le *XUI Vendor Shell Widget Class*
  - -- Si une application fait son édition de lien avec Motif, on utilise le *Motif Vendor Shell Widget Class*
  - -- Les *widgets* Motif requièrent un *Motif Vendor Class Shell*

- -- Les *widgets* XUI sont compatibles avec le *Motif Vendor Class Shell*

- Le problème n'existe donc que si une application XUI active dynamiquement (avec LIB$FIND_IMAGE_SYMBOL) une image partageable utilisant Motif
  - -- Dans ce cas, le *Toolkit* prend la décision d'utiliser le *XUI Vendor Shell* lors de son initialisation et les *Widgets* Motif dans l'image partageable activée dynamiquement ne fonctionnent pas
  - -- La méthode de contournement est d'ajouter le nom de l'application au nom logique DECW$USE_XM_VENDOR_SHELL
  - -- Ce nom logique contient sous la forme d'une *search-list* la liste des noms d'images pour lesquels le *Toolkit* doit utiliser le *Motif Vendor Shell* lors de son initialisation

- -- La valeur par défaut de ce nom logique est NOTES$MAIN
- -- Il faut donc définir le nom logique sous la forme : "NOTES$MAIN,IMAGE1,IMAGE2,..."

- Il peut être nécessaire de modifier les procédures d'éditions de lien pour les applications XUI :
  - -- Il faut inclure explicitement SYS$SHARE:DECW$XLIBSHR.EXE
  - -- En effet, DECW$DWTLIBSHR.EXE n'est plus lié à DECW$XLIBSHR.EXE
  - -- Si la procédure d'édition de lien ne contenait que la ligne :
    DECW$DWTLIBSHR.EXE/SHARE
    il faut ajouter la ligne :
    DECW$XLIBSHR.EXE/SHARE

Nouvelles fonctionnalités

Le Toolkit MOTIF

La valeur ajoutée de DIGITAL :

- DXM : DIGITAL eXtension to Motif :
  - -- *Color Mix Widget*
  - -- *Compound String Text Widget*
  - -- *Help Widget*
  - -- *Print Widget*
  - -- *Scructured Visual Navigation Widget*
- Outils de portage de XUI à Motif
- Différents *bindings* (C, Ada, FORTRAN, Pascal)
- Amélioration de performances par rapport à OSF/Motif
- Beaucoup de *bugs* fixés par rapport à OSF/Motif

Nouvelles fonctionnalités

Les applications portées vers Motif

- Bookreader
- CDA Viewer
- Calculator
- Calendar
- Cardfiler
- Clock
- DECterm
- FileView
- Mail
- Paint
- Session Manager

Nouvelles fonctionnalités

Les applications non portées vers Motif

- Print Screen
- Notepad

Nouvelles fonctionnalités

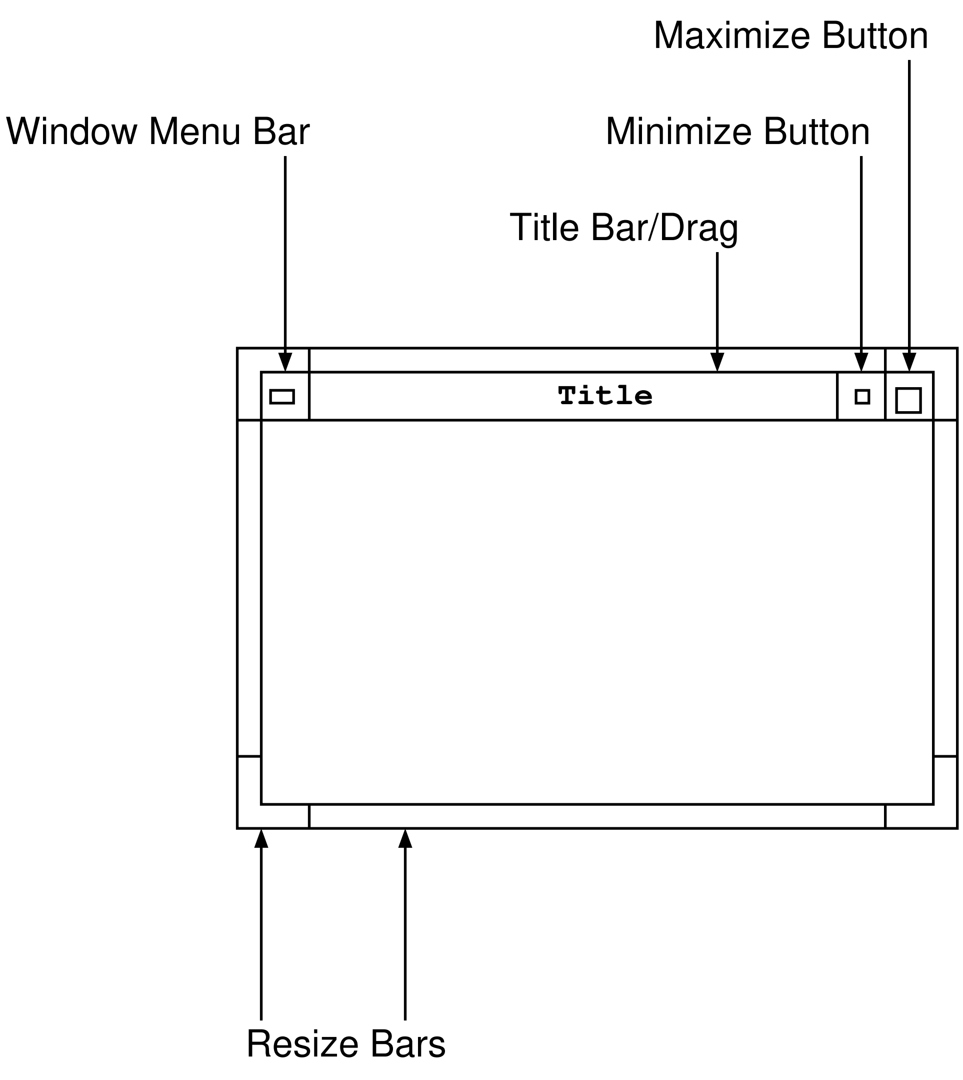
Le Window Manager Motif

- Motif est le *Window Manager* par défaut
- Le *window Manager* XUI est aussi disponible
- *"Look"* 3-D
- Permet de très nombreuses *"customizations"*
  - -- menus
  - -- buttons
  - -- keys
  - -- color
  - -- focus
  - -- icons
  - -- icon box

- Suit les recommandations contenues dans l'*Inter-Client Communications Conventions Manual* (ICCCM)

Nouvelles fonctionnalités

Les décorations du Window Manager Motif



Nouvelles fonctionnalités

Le Window Manager Motif (Mwm)

- DECwindows Motif inclut les fichiers de *resource* du *Window Manager* Motif permettant de modifier le comportement et l'apparence du *Window Manager*
- Ces fichiers sont les suivants :
  - -- Fichier de *resources* couleur :
    - > les fichiers par défaut sont situés dans la *directory* pointée par le nom logique DECW$SYSTEM_DEFAULTS
    - > DECW$MWM.DAT : contient toutes les *resources* du *Window Manager*
    - > DECW$MWM_BW.DAT : contient les *resources* du *Window Manager* pour les stations noir & blanc

> DECW$MWM_COLOR.DAT : contient les *resources* du *Window Manager* pour les stations couleur

> DECW$MWM_GRAY.DAT : contient les *resources* du *Window Manager* pour les stations multiplans noir & blanc

-- Fichier de configuration :

> le fichier par défaut est le fichier DECW$SYSTEM_DEFAULTS:DECW$MWM_RC.DAT

> contient la définition des menus du *Window Manager* et le *mapping* du clavier

- Comment Mwm cherche ses *resources* ? :

-- Mwm cherche successivement dans les *directories* DECW$USER_DEFAULTS et DECW$SYSTEM_DEFAULTS le fichier DECW$MWM.DAT

-- Pour les moniteurs monochromes noir & blanc et multiplans, Mwm supperpose les définitions des *resources* couleur définies dans DECW$MWM.DAT par celles définies dans DECW$MWM_BW.DAT et dans DECW$MWM_GRAY.DAT

-- Une *resource* définie dans le fichier Xdefaults supperpose toute *resource* contenue dans le fichier DECW$MWM.DAT

-- Pour utiliser les nouvelles *resources* définies dans les fichiers Mwm, il faut redémarrer le *Window Manager* depuis le *Workspace Menu* (activé en pressant MB1 ou MB3 dans la *root window*)

-- Pour utiliser les nouvelles *resources* définies dans le fichier Xdefault, il faut sortir du *Session Manager* et se reconnecter

-- Il est donc préférable de placer ses *resources* dans les fichiers Mwm plutôt que dans le fichier Xdefault

- Le fichier de configuration DECW$MWM_RC.DAT définit la plupart des utilisations par le *Window Manager* des touches de fonction :
  - -- si une application utilise ces touches de fonction, il faut commenter la ligne correspondante dans le fichier DECW$MWM_RC.DAT en plaçant au début de la ligne un commentaire (# ou !)
  - -- le *binding* Motif "Alt (compose) espace" a été commenté afin de ne pas interférer avec les séquences composées du DECterm, de DECwrite, DECpresent
  - -- F11 (*shift escape*) fait apparaître le *root menu*
- Le fichier des *bindings* Motif qui permet de faire la correspondance entre les *events* X et les actions sur les boutons, clés ou touches est dans le fichier DECW$USER_DEFAULTS:DECW$MOTIFBIND.DAT

- La *directory* de recherche des *bitmaps* est la directory DECW$USER_DEFAULTS; cette *directory* peut être redéfinie en spéficiant la *resource* : Mwm*bitmapDirectory
- Le fichier SYS$MANAGER:DECW$MWM.COM est celui qui est utilisé pour redémarrer le *Window Manager*

La flexibilité du Session Manager

- Nouvelles fonctionnalités *FileView-like*
- Fonctions *Session Manager* appelables depuis n’importe quel langage
- Fichier de commande de paramètrage d’environnement
- Fichiers *Profile*

La flexibilité du Session Manager

Nouvelles fonctionnalités FileView-like

- Le *Session Manager* est une barre de menu complètement *customizable*
- *Customization* possible de cette barre de menu et des choix du menu en utilisant une méthode de type *FileView*
- Cette *Customization* est faite en faisant un "merge" des fichiers *profile*
- Les applications sont suivies à travers une boite de dialogue de type *Work-In-Progress*

La flexibilité du Session Manager

Fonctions Session Manager appelables

- Les fonctions du *Session Manager* sont des images partageables appelées dynamiquement :
  - -- Pause Session :
    - > par défaut : Sys$System:DECW$PAUSESESSION.EXE
    - > *customisation* dans le fichier **DECW$USER_DEFAULTS:DECW$PAUSESESSION.DAT**
    - > accessible depuis les menus du *Session Manager*
  - -- End Session :
    - > par défaut : Sys$System:DECW$ENDSESSION.EXE
    - > *customisation* dans le fichier **DECW$USER_DEFAULTS:DECW$ENDSESSION.DAT**

    - > accessible depuis les menus du *Session Manager*
- -- Message Panel :
    - > par défaut : Sys$System:DECW$MESSAGEPANEL.EXE
    - > *customisation* dans le fichier **DECW$USER_DEFAULTS:DECW$MESSAGEPANEL .DAT**
    - > accessible depuis les menus du *Session Manager*

- Les fichiers de *customisation* principaux du *Session Manager* sont :
    - -- **DECW$USER_DEFAULTS:DECW$SMB_GENERAL.DAT**
    - -- **DECW$USER_DEFAULTS:DECW$SMB_COLOR.DAT**
    - -- **DECW$USER_DEFAULTS:DECW$SMB_GRAY.DAT**
    - -- **DECW$USER_DEFAULTS:DECW$SMB_BW.DAT**
    - -- **DECW$USER_DEFAULTS:DECW$SMB_BACKGROUND.DAT**
    - -- **DECW$USER_DEFAULTS:DECW$SMB_BACKGROUND_BW.DAT**
    - -- **DECW$USER_DEFAULTS:DECW$SMB_BACKGROUND_COLOR.DAT**

- -- DECW$USER_DEFAULTS:DECW$SMB_BACKGROUND_GRAY.DAT
- -- DECW$USER_DEFAULTS:DECW$SMB_KEYBOARD.DAT
- -- DECW$USER_DEFAULTS:DECW$SMB_LANGUAGE.DAT
- -- DECW$USER_DEFAULTS:DECW$SMB_POINTER.DAT
- -- DECW$USER_DEFAULTS:DECW$SMB_POINTER_BW.DAT
- -- DECW$USER_DEFAULTS:DECW$SMB_POINTER_COLOR.DAT
- -- DECW$USER_DEFAULTS:DECW$SMB_POINTER_GRAY.DAT
- -- DECW$USER_DEFAULTS:DECW$SMB_SECURITY.DAT
- -- DECW$USER_DEFAULTS:DECW$SMB_WINDOW.DAT
- -- DECW$USER_DEFAULTS:DECW$SMB_WINDOW_BW.DAT
- -- DECW$USER_DEFAULTS:DECW$SMB_WINDOW_COLOR.DAT
- -- DECW$USER_DEFAULTS:DECW$SMB_WINDOW_GRAY.DAT

- Si l'on utilise un **autre** *Session Manager* il est quand même possible d'avoir les fonction-alités de la *Session*
- Très flexible

La flexibilité du Session Manager

Fichier de commande de paramètrage d'environnement

- Il est possible de *customizer* l'environnement DECwindows avec : sys$manager:decw$private_apps_setup.template
  - -- Semblable à decw$private_server_setup.com
  - -- Copier dans un fichier .com et editer
- On peut modifier :
  - -- **La *directory* du *bookreader*** :
    - > **Défaut : sys$common:[decw$book]**
    - > **Symbole : DECW$BOOK**
  - -- **La *directory* des fichiers de commandes *FileView* :**
    - > **Défaut : decw$syscommon:[vue$library.user],**
      decw$syscommon:[vue$library.system]
    - > **Symbole : VUE$LIBRARY**

- **La *directory* des fichiers *Keymap* :**
    - **> Défaut : decw$syscommon:[sys$keymap.decw.user], decw$syscommon:[sys$keymap.decw.system]**
    - **> Symbole : DECW$KEYMAP**
- **La *directory* des fichiers d’exemples DECwindows :**
    - **> Défaut : sys$common:[syshlp.examples.decw]**
    - **> Symbole : DECW$EXAMPLES**
- **La *directory* des fichiers d’exemples XDPS :**
    - **> Défaut : sys$common:[syshlp.examples.xdps]**
    - **> Symbole : XDPS$EXAMPLES**
- **La *directory* pour les fichiers de commandes de barres de menus du *Session Manager* :**
    - **> Défaut : pas de défaut**
    - **> Symbole : VUE$MBLIBRARY**
- **Le programme qui s’exécute quand la séquence de démarrage de DECwindows est terminée :**
    - **> Défaut : decw$startlogin**
    - **> Symbole : DECW$MAINAPP**

- -- **Le logo utilisé pendant *loginout* :**
  - > **Défaut : pas de défaut**
  - > **Symbole : DECW$LOGINLOGO**
- -- **Les paramètres par défaut de la commande SET DISPLAY :**
  - > **Défaut : node 0, server 0, screen 0**
  - > **Symboles : DECW$APPSNODE, DECW$APPSSERVER, DECW$APPSSCREEN**
- -- **La procédure de commande qui s’exécute après la procédure d’autorisation (doit démarrer un session manager) :**
  - > **Défaut : sys$manager:decw$startsm.com**
  - > **Symbole : DECW$SESSIONCOM**
- -- **La procédure de sylogin DECwindows :**
  - > **Défaut : pas de défaut**
  - > **Symbole : DECW$SYLOGINCOM**
- -- **La commande qui initialise la station avec les bonnes valeurs de *resources* utilisateur :**
  - > **Défaut : run sys$system:decw$wsinit**
  - > **Symbole : DECW$SESSIONINIT**

-- **La commande qui exécute une session :**

  > **Défaut : run sys$system:decw$session**

  > **Symbole : DECW$SESSIONMAIN**

-- **La commande qui termine une session**

  > **Défaut : pas de défaut**

  > **Symbole : DECW$SESSIONEND**

-- **Le fichier "log" pour loginout :**

  > **Défaut : pas de défaut**

  > **Symbole : DECW$LOGINLOG**

-- **La permission d’autoriser plusieurs connections pendant loginout**

  > **Défaut : non**

  > **Symbole : DECW$LOGINMANY**

-- **Supprimer le fichier "log" du *Session Manager* :**

  > **Défaut : non**

  > **Symbole : DECW$SESSIONLOG**

- Exemple: il est possible de spécifier l’*executable* du *session manager* :
  - -- Definir le nom logique permettant de paramètrer l’*executable* du *session manager* pour pointer sur un fichier de commande contenant la ligne :

    run sys$manager:decw$bookreader

  - -- Définir le nom logique permettant de terminer une *session* pour être "run decw$endsession"
  - -- Quand l’utilisateur se connecte, *Bookreader* démarre. Quand il quitte *Bookreader,* la *session* prend fin

La flexibilité du Session Manager

Compte dédié

- Il était très difficile de créer des comptes dédiés sous DECwindows : DECwindows MOTIF V1.0 permet de le faire
- En voici 3 exemples :
  - -- Modification de la procédure de commande du *Session Manager*
  - -- Modification de l'*executable* du *Session Manager*
  - -- Modification du fichier profil du *Session Manager*

Modification de la procédure de commande du *Session Manager* :

- Il faut définir le symbole global DECW$SESSIONCOM dans le fichier SYS$MANAGER:DECW$PRIVATE_APPS_SETUP.COM
- Dans ce cas, la procédure SYS$MANAGER:DECW$STARTAPPS.COM définit le nom logique DECW$SESSIONCOM dans la table de noms logiques système
- Ceci affecte donc le démarrage du *Session Manager* pour **tous** les utilisateurs
- Il faut donc dans sa procédure ”privée” de démarrage du *Session Manager* tester le nom de l’utilisateur concerné
- Avec cette méthode, les procédures de commande qui s’exécutent normalement depuis DECW$STARTSM.COM comme SYLOGIN.COM, LOGIN.COM, DECW$SYLOGIN.COM et DECW$LOGIN.COM ne sont **pas exécutées**

Exemple de procédure :

```
$!++
$!  Check to see whether this is the dedicated DECwindows account.
$!   If it isn't, run the normal procedure to start the Session Manager
$!--
$ username = F$USER()
$ IF F$LOCATE("BOOK_READER", username) .NES. -
      F$LENGTH(username) THEN GOTO dedicated
$!+
$!  Normal, nondedicated DECwindows login
$!-
$ @SYS$MANAGER:DECW$STARTSM
$ EXIT
$dedicated:
$!
$ RUN SYS$SYSTEM:DECW$WSINIT
$ display = F$LOGICAL("decw$display")
$ RUN/DETACHED/OUTPUT='display' SYS$SYSTEM:DECW$MWM
$ RUN SYS$SYSTEM:DECW$BOOKREADER
$ endsession := $DECW$ENDSESSION
$ endsession -noprompt
$ STOP/ID=0
```

- Il faut ensuite definir le symbole :

DECW$SESSIONCOM:==SYS$MANAGER:PRIVATE_SESSIONCOM.COM

dans la procédure :

SYS$MANAGER:DECW$PRIVATE_APPS_SETUP.COM

- Il faut enfin redémarrer DECwindows par :

@SYS$MANAGER:DECW$STARTUP RESTART

Modification de l'*executable* du *Session Manager* :

- Il faut définir le symbole global DECW$SESSIONMAIN dans le fichier SYS$MANAGER:DECW$PRIVATE_APPS_SETUP.COM
- Dans ce cas ,la procédure SYS$MANAGER:DECW$STARTSM.COM exécute la procédure de commande spécifiée au lieu de DECW$SESSION.EXE
- Le *Session Manager* lit les ficihers de *re-sources* et **exécute les procédures de login standards** mais ne montre pas la barre de menu du *Session Manager*
- Ceci affecte donc le démarrage du *Session Manager* pour tous les utilisateurs
- Il faut donc dans sa procédure ”privée” de démarrage du *Session Manager* tester le nom de l'utilisateur concerné

Exemple de procédure :

```
$!++
$!  Check to see whether this is the dedicated DECwindows account.
$!   If it isn't, run the normal procedure to start the Session Manager
$!--
$ username = F$USER()
$ IF F$LOCATE("BOOK_READER", username) .NES. -
     F$LENGTH(username) THEN GOTO dedicated
$!+
$!  Normal, nondedicated DECwindows login
$!-
$ RUN SYS$SYSTEM:DECW$VUEMASTER
$ EXIT
$dedicated:
$!
$ display = F$LOGICAL("decw$display")
$ RUN/DETACHED/OUTPUT='display' SYS$SYSTEM:DECW$MWM
$ RUN SYS$SYSTEM:DECW$BOOKREADER
$ endsession := $DECW$ENDSESSION
$ endsession -noprompt
$ STOP/ID=0
```

- Il faut ensuite definir le symbole :

DECW$SESSIONMAIN:==SYS$MANAGER:PRIVATE_SESSIONMN.COM

dans la procédure :

SYS$MANAGER:DECW$PRIVATE_APPS_SETUP.COM

- Il faut enfin redémarrer DECwindows par :

@SYS$MANAGER:DECW$STARTUP RESTART

Modification du fichier profil du *Session Manager* :

- Il est possible de laisser démarrer le *Session Manager* normalement et de le *customiser* afin d'interdire à l'utilisateur d'utiliser toute commande non désirée
- Il faut pour cela créer le compte dédié de l'utilisateur, se connecter sous ce compte, utiliser les barres de menus du *Session Manager* pour supprimer les commandes non désirées et sauvegarder le profil correspondant
- Le fichier profil est sauvegardé dans le fichier :
  decw$user_defaults:VUE$PROFILE.VUE$DAT

La flexibilité du Session Manager

Les Fichiers profil du Session Manager

- Le *Session Manager* utilise des fichiers profil (*profile file*) de type *FileView* :
  - -- System profile files
  - -- Public profile files
  - -- User profile files
- Pour éviter les confusions, le *Session Manager* **et** le *FileView* utilisent le **même fichier profil** :
  decw$user_defaults:VUE$PROFILE.VUE$DAT
  - -- Création d'un nouveau *startup view* pour le *Session Manager* appelé *Session Manager* : le *startup view* pour le *FileView* garde le même nom: *Startup*

- -- Les options du menu servent à la fois pour le *FileView* et le *Session Manager*
- -- L'utilisateur peut cependant définir 2 menus avec les même noms de barres de menus MAIS avec différents choix : par exemple, le menu d'options du *FileView* est différent de celui du *SessionManager*

- Le *System Manager* à la possibilité de paramètrer la barre de menu du *Session Manager* pour ne contenir que les choix désirés :
  - -- Pour tout le système
  - -- Pour des groupes par UIC par exemple
- On peut supprimer les *customization menus*

La séquence de Startup

Par défaut la séquence suivante est exécutée quand un utilisateur de connecte :

1. *Loginout* et *DECW$Loginout* travaillent ensemble pour obtenir un couple *username/password* valide
2. *Loginout* modifie la *security list* pour autoriser l'utilisateur connecté à créer des liens entre les clients et le serveur DECwindows
3. *Loginout* démarre un *process* qui maintient la connexion avec le serveur pendant le démarrage de la session (ce *process* disparait ensuite) afin de permettre au *window manager* (DECwindows ou Motif) de se connecter au serveur
4. *DECW$Loginout* crée le *window manager* (*process* DECW$MWM pour le *window manager* Motif)

5. *Loginout* rend la main au DCL qui exécute ses commandes avec SYS$INPUT pointant sur le fichier de commande du *Session Manager.* Par défaut, ce fichier est SYS$MANAGER:DECW$STARTSM.COM
6. Exécution de SYLOGIN.COM
7. Exécution de LOGIN.COM : l'utilisateur peut redéfinir DECW$USER_DEFAULTS à ce moment là
8. Exécution de l'initialisation de la station en fonction des fichiers de *resources*. Par défaut l'initialisation est effectuée par SYS$SYSTEM:DECW$WSINIT.EXE
9. Si c'est la première fois que l'utilisateur se connecte sous DECwindows Motif, les nouveaux fichiers de *resources* sont créés
10. Exécution de DECW$SYLOGIN.COM
11. Exécution de DECW$LOGIN.COM

12. Exécution du *Session Manager*. Par défaut, le fichier exécuté est SYS$SYSTEM:DECW$SESSION.EXE

13. Exécution de l'*end session*. Par défaut, le fichier exécutable d'*end session* est SYS$SYSTEM:DECW$ENDSESSION.EXE

Remarques importantes :

- Pendant le démarrage du *Session Manager*, le fichier de *resource* utilisateur DECW$XDEFAULTS.DAT et tous les fichiers *resource* de la session sont lus depuis la *directory* DECW$USER_DEFAULTS et pas depuis le *directory* SYS$LOGIN comme c'était le cas dans le passé : par défaut DECW$USER_DEFAULTS pointe sur SYS$LOGIN. En cas de problème, DECW$USER_DEFAULTS peut être redéfini dans l'un des fichiers DECW$LOGIN.COM ou LOGIN.COM
- Cette version de DECwindows permet de fixer les problèmes de la VT1300 (*end session* et *start session*). Pour

bénéficier de cette correction, il faut éditer le fichier EWS$LIBRARY:EWS$SERVER.COM et ajouter la ligne :

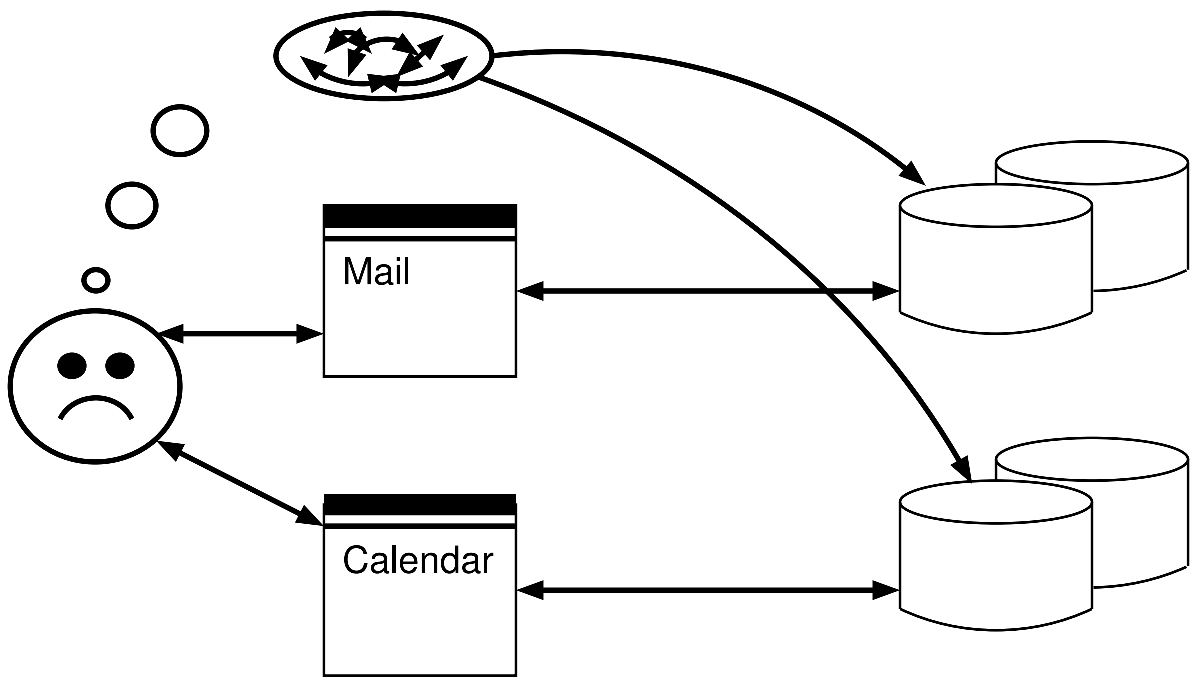
$DEFINE DECW$LOGIN_RESTART TRUE

L’hyperInformation

- Le problème
- La solution
- Les applications

L'hyperInformation

Le problème



L'hyperInformation

La solution

L'utilisation de l'HyperInformation

Un peu d'histoire.... :

- L'HyperInformation est une technologie dérivée de celle d'Hypertext
- La notion d'Hypertext a été introduite par un visionnaire en 1967 : Ted Nelson inspiré par le rève de Vannevar Bush qui était le conseiller scientifique de F.D. Roosevelt et qui cherchait à l'aide des technologies de l'époque (les biblio-thèques de microfilms) à résoudre les problèmes de complexité d'accès aux documentations scientifiques

- La notion d'Hypertext a été la première tentative de résolution de ce problème en permettant une approche non linéaire à l'écriture des documents :
  - -- notion d'association d'idées
  - -- création de références à l'intérieur du document
  - -- le lecteur devient le créateur de l'information et non simplement un consommateur
- Les 2 limitations de la notion d'Hypertext sont liées au fait que :
  - -- cette notion ne concerne que les documents (pas d'image, de son, de multi-media)
  - -- cette notion est isolée : il faut faire un "effort" pour intégrer l'information dans le système

- Le but de la notion de l'HyperInformation est de résoudre ces 2 problèmes en créant un partenariat entre :
  - -- Les applications
  - -- Les données des applications
  - -- Les services d'HyperInformation sous-jacents
- DECwindows Motif V1.0 introduit cette notion sous le nom de LinkWork au moyen de 2 composants :
  - -- Les services d'HyperInformations :
    - > ensemble de services permettant aux applications de participer à l'environnement d'HyperInformation
    - > librairie partageable : SYS$SHARE:LWK$DXMSHR.EXE
  - -- Les services d'HyperSession :
    - > LinkWorks Environment Manager :

+ permet de lister, ajouter, suprimer des objets de l’environnement d’HyperInformation
+ permet de contrôler le fonctionnement par défaut des hyperapplications
+ SYS$SYSTEM:DECW$LWK_MANAGER.EXE

> LinkWorks Setup Utility :

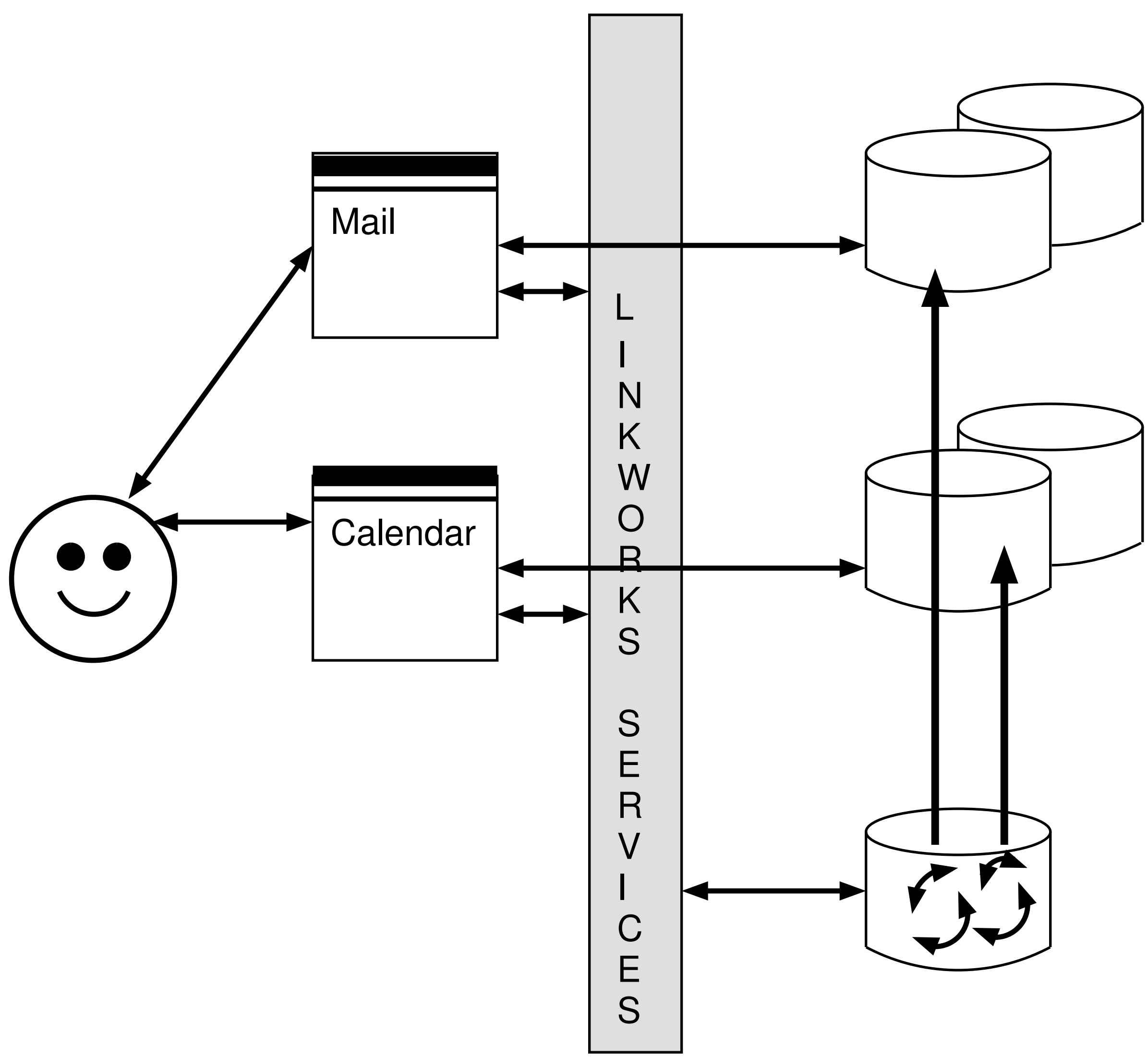
+ sous-ensemble de l’*environment manager*
+ démarré par défaut par le *Session Manager* (SYS$SYSTEM:DECW$LWK_SETUP.EXE)
+ indispensable pour utiliser toute hyperapplication

Comment cà marche ? :

- Les applications procurent les informations sur les objets

- L’utilisateur crée des liens entre les objets

- Par exemple, il est possible de créer un lien entre un entrée dans le calendrier et un message *mail*
- L’HyperSession gère l’HyperEnvironment



- Applications supportant initialement la notion d'HyperInformation :
  - -- Bookreader
  - -- Calendar
  - -- Cardfiler
  - -- Mail
- Les informations au sujet des liens établis par l'utilisateur sont stockés dans une *hyperinformation* ***linkbase***
  - -- SYS$LOGIN:LWK_PERSONAL.LINKBASE
  - -- Pointé par le nom logique LWK_PERSONAL
  - -- Il est possible de créer plusieurs *linkbases*
  - -- Cette base de donnée contient :
    - > des *linknets* (information sur le lien)
    - > des *linknet lists* (ensemble de *linknets*)
    - > des *pathes* et des *pathes lists* (historique des liens qui ont été créés)

Divers

- Améliorations du DECwindows *server*
- Fonctionnalités additionnelles pour les applications
- Améliorations de qualité
- Placement des *fonts*
- Cours Motif assisté par ordinateur

Divers

Améliorations du DECwindows server

- Amélioration de performances lors des opérations de fenêtrage (voir plus loin)
- Support du ***Backing Store*** :
  - -- Autorise le serveur à sauvegarder le contenu d'une fenêtre recouverte dans sa mémoire d'arrière plan
  - -- Permet d'éviter de recréer la fenêtre quand elle n'est plus recouverte
- Compatible MIT X11 R4
- *Directory* de fichiers *fonts* pour améliorer les performances du *startup*
- Support d'un *font alias file* pour améliorer l'interopérabilité (voir plus haut)

- Ajout de nouvelles extensions :
  - -- X Imaging Extension (XIE)
  - -- Phigs Extension for X (PEX) pour le *hardware* SPX
- Display PostScript© bitmap glyphs : améliore considérablement le rendu sur l'écran
- Support de nouveau *hardware*

Divers

Fonctionnalités additionnelles pour les applications

- *Computer Based Instruction* pour un nouvel utilisateur Motif
- Support de DAS (DECimage Application Services) V3.0
- Paint supporte complètement la couleur
- *Bookreader* permet à plusieurs livres et à plusieurs sujets d'être ouverts en même temps
- *Calculator* calcule en hexadécimal et en octal et supporte les entrées au clavier

Divers

Améliorations de qualité

- La qualité du *Server*, de la *Xlib* et du *Toolkit* XUI est meilleure que celle de DECwindows V2
- Motif est un produit en V1 en provenance de l'OSF
- Le *Server,* la Xlib et le *toolkit* ont un certain nombre d'améliorations de performance (voir plus loin)
- Les performances de Motif seront moins bonnes que celles de XUI (voir plus loin)
- La plupart des problèmes de performance de Motif sont dues à des fonctionnalités additionnelles de Motif par rapport à XUI comme la notion de *keyboard traversal* (voir plus loin)

Divers

Placement des fonts

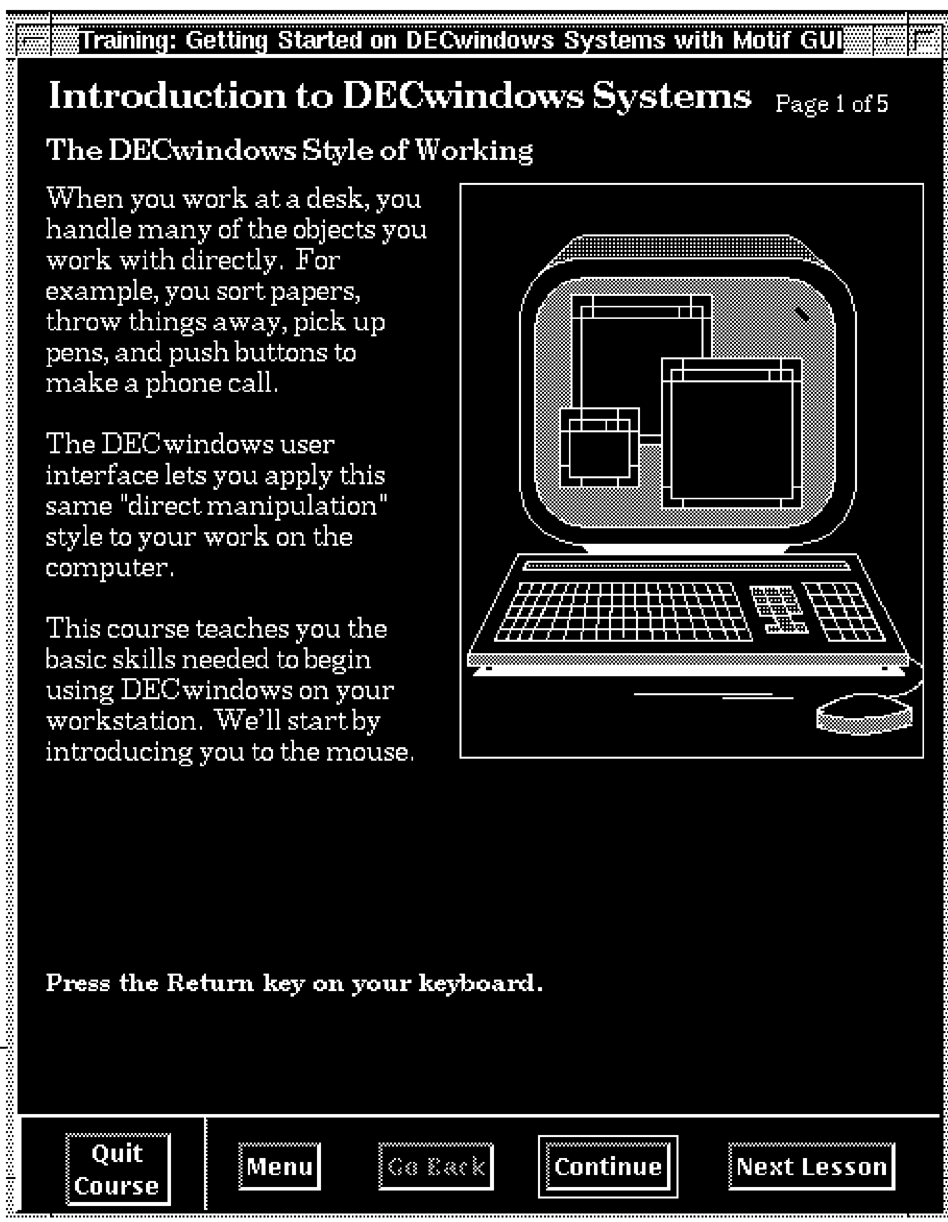
- Les *fonts* "utilisateurs" doivent être placées dans les nouvelles *directories* réservées pour cela :
  - -- SYS$COMMON:[SYSFONT.DECW.USER_75DPI]
  - -- SYS$COMMON:[SYSFONT.DECW.USER_100DPI]
  - -- SYS$COMMON:[SYSFONT.DECW.USER_COMMON]
  - -- SYS$COMMON:[SYSFONT.DECW.USER_CURSOR16]
  - -- SYS$COMMON:[SYSFONT.DECW.USER_CURSOR32]
- Les autres *directories* sont réservées aux *fonts* distribuées avec DECwindows
- Attention à DECW$MOTIF_TAILOR qui peut supprimer des *fonts* si celles-ce sont placées dans les mauvaises *directories*
- Il est possible d'utiliser des *font alias files* pour appeler différentes *fonts* avec des noms

identiques : pour cela , il faut créer un fichier *font alias* placé dans la même *directory* que la *font* qu'il renomme avec un nom de fichier DECW$FONT_ALIAS*.DAT

- Si l'on installe une nouvelle *font sur le système*, il faut exécuter le fichier : SYS$UPDATE:DECW$MKFONTDIR.COM pour "installer" correctement la *font* (cela permet de l'entrer dans le(s) fichier(s) : DECW$FONT_DIRECTORY*.DAT

Divers
Cours MOTIF

- Invoqué par $RUN SYS$SYSTEM:DECW$CBI
- Utilise les fichiers pointés par le nom logique DECW$CBI_SUPPORT



Le nouveau Packaging

- DECwindows va devenir un *layered product* appelé VMS DECwindows Motif V1.0 (appelé précédemment DECwindows V3.0)
- Ce nouveau produit sera être disponible à la rentrée comprenant :
  - -- le support de la partie serveur V3.0 (support de nouveau *hardware)*
  - -- le support de la partie client V3.0
- Les prochaines versions de VMS (V5.4-3 et V5.5) seront livrées avec :
  - -- le support de la partie serveur V3.0 (support de nouveau *hardware)*
  - -- le support de la partie client V2.1

- Cette modification permet :
  - -- de séparer les versions de VMS et de DECwindows afin de pouvoir délivrer plus vite les nouvelles versions produites par l'OSF
  - -- de ne pas pénaliser l'ensemble des utilisateurs de VMS des royalties payées à l'OSF et à ADOBE
- VMS DECwindows Motif V1.0 est supportée à partir de VMS V5.4
- A terme, VMS ne contiendra que les fichiers correspondant au code du serveur DECwindows

Les performances

Les résultats de performances présentés ici sont des résultats basés sur une version *field test* et doivent donc être considérés avec précaution

- Les fonctionnalités d'*autotuning* disponibles à partir de VMS V5.4-3 permettent à DECwindows Motif V1.0 de tourner sans problème avec 8 Mbytes

- Le pourcentage de mémoire dynamique consommée en plus par Motif par rapport à XUi est de 20% à 30%. Ceci est dû à :
  - -- l'accroissement de la taille des *widgets* Motif par rapport à XUI
  - -- l'accroissement de la taille des *gadgets* Motif par rapport à XUI malgré le *gadget caching*
  - -- l'accroissement du nombre de structures de données nécessaires pour supporter la notion de *keyboard traversal* (permettant, entre autres choses, de spécifier qu'un *widget* accepte ou non le *focus*)

- La consommation CPU a également augmenté
- Exemple des performances obtenues sur une VS2000 (standalone, 6 Mbytes, RD54, monochrome, VMS V5.4) :
  - -- Temps de login (démarrage d’un FileView et d’un DECterm) :
    - > V2 : 63 s
    - > V3 : 135 s
  - -- Temps de démarrage d’un DECterm :
    - > V2, premier DECterm : 22 s
    - > V2, deuxième DECterm : 16 s
    - > V3, premier DECterm : 29 s
    - > V3, deuxième DECterm : 22 s

-- Temps de Pause :
  > V2, Pause : .5 s
  > V2, Un-pause : 2 s
  > V3, Pause : 2 s
  > V3, Un-pause : 2.5 s
-- *Logout* :
  > V2, Fin de session -> boite de dialogue : 2 s
  > V2, Boite de dialogue -> Logo : 33 s
  > V3, Fin de session -> boite de dialogue : 6 s
  > V2, Boite de dialogue -> Logo : 45 s
-- Démarrage du *FileView* :
  > V2, depuis le *Session Manager* : 16 s
  > V3, depuis le *Session Manager* : 7 s

- Performance du serveur DECwindows par rapport à celui du MIT (X11R4) en utilisant X11perf sur une VS2000 (dans les conditions indiquées ci-dessus) :

Opération	MIT R4	DECWindows
1*1 rectangle	4220	7400
500*500 rectangle	56.5	82.7
10-pixel line segment	3510	5280
500-pixel line segment	244	519
10-pixel dashed segment	2540	393
10*1 wide line	469	175
PutImage 100*100 square	56.9	89.1
PutImage 500*500 square	4.7	9.9
GetImage 100*100 square	47.5	55
GetImage 500*500 square	4.6	11.3
Copy 100*100 window to pix	145	206
Copy 100*100 pix to window	150	197
......		

- D’une façon générale, les performances du serveur de DECwindows sont supérieures à celles sur serveur X11R4 du MIT à l’exception des tracés de lignes en pointillé ou du tracé d’ellipses

Questions