



UNIVERSITÉ  
PARIS  
DESCARTES

# IHM-BD et Ergonomie

DUT1  
Mme HEIWY  
*Janvier 2018*

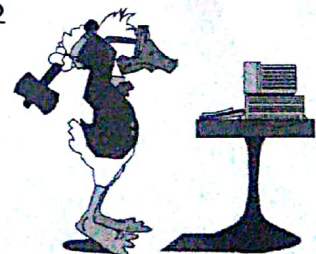
# Organisation du cours IHM

- ♣ IHM 1<sup>ere</sup> partie
- ♣ Durée du cours : Période C (7 semaines)
- ♣ Responsable du cours: Mme Mesatfa-Fessy
- ♣ Volumétrie du cours:
  - ♦ 1 CM / semaine
  - ♦ 1 TP de 1h30/ semaine
- ♣ Modalités d'évaluation prévue: Projet IHM

2

## LES IHM (INTERFACES HOMME-MACHINE) et l'ergonomie: *Plan*

- ♣ INTRODUCTION & DEFINITION
- ♣ LES ETAPES DE CREATION D'UNE INTERFACE GRAPHIQUE
- ♣ LES TYPES DE DEVELOPPEMENT d'IHM
- ♣ LA PROGRAMMATION EVENEMENTIELLE
- ♣ DEVELOPPEMENT D'UNE INTERFACE AVEC ACCESS
  - ♦ CREATION DE FORMULAIRE SANS MACRO
  - ♦ CREATION DE FORMULAIRE AVEC MACRO
  - ♦ DEVELOPPEMENT VBA
- ♣ L'ERGONOMIE
- ♣ CONCEPTION D'INTERFACE GRAPHIQUE
- ♣ MAQUETTAGE ET ENCHAINEMENT DES FENETRES
- ♣ DIAGRAMME UML POUR REPRESENTER LES ENCHAINEMENTS
  - ♣ DIAGRAMME D'ETATS
  - ♣ DIAGRAMME DE COLLABORATION



3

# INTRODUCTION & DEFINITION

## ♣ A quoi sert une interface?

- ♦ A permettre l'utilisation des fonctionnalités à des utilisateurs qui n'ont pas forcément les connaissances informatiques du concepteur du logiciel;
- ♦ A faciliter l'utilisation des fonctionnalités du logiciel.

## ♣ Que contient une interface graphique?

- ♦ Un ensemble d'objets graphiques (menu, boutons, fenêtres, etc.) permettant d'accéder aux fonctionnalités de l'application.

© V. Heiwy

4

# LES ETAPES DE CREATION D'UNE INTERFACE GRAPHIQUE

## ♣ Comment réaliser l'interface de la future application?

### 1. Concevoir l'interface :

- ♥ A partir de spécifications fonctionnelles (MOT (Modèle Organisationnel des Traitements) de MERISE (vois S4) ou DCU (diagrammes de cas d'utilisation) UML), réaliser le **maquettage** i.e. dessiner des écrans attendus
  - ▲ Traitements interactifs => Ecran de saisie (Formulaire)
  - ▲ Traitements automatiques => Élément du menu (Editions, etc.)
- ♥ Définir l'**enchaînement** entre les fenêtres

### 2. Développer l'interface en utilisant la programmation événementielle

© V. Heiwy

5

# Programmation événementielle

- ♣ En informatique, la programmation événementielle est un paradigme de programmation fondé sur les événements.
- ♣ Elle s'oppose à la programmation séquentielle.
- ♣ Le programme se définit par ses réactions aux événements qui peuvent se produire, c'est-à-dire des changements d'état de variable.
- ♣ Exemple d'événements: l'incrément d'une liste, un mouvement de souris ou de clavier.

(wikipedia)

- ♣ La survenance d'un événement (clic sur un bouton, etc.) déclenche des actions, i.e. un programme

6

## Les types de développement d'Interfaces

- ♣ Pour développer une interface graphique c'est possible:
  - ♦ Dans un contexte de développement classique, par exemple en VB (cf. cours J. Rossit)
  - ♦ Dans le cadre de développement Web: avec Javascript et AJAX côté client pour développer des formulaires ou JQuery; Node.js (côté serveur)
  - ♦ Dans un contexte développement orienté-objet, par exemple en JAVA, grâce à l'utilisation de la bibliothèque graphique SWING
  - ♦ Dans un contexte « base de données », par exemple avec ACCESS: en créant des formulaires avec ou sans macro ou en développant des modules VBA
  - ♦ Dans le cadre d'applications mobiles aussi (SDK Android, Framework jQuery mobile, Framework Sencha Touch, etc.) (cf. semestre 4)

7

# DEVELOPPEMENT D'INTERFACE AVEC ACCESS

©V. Heiwy

8

## Développement d'interface avec ACCESS

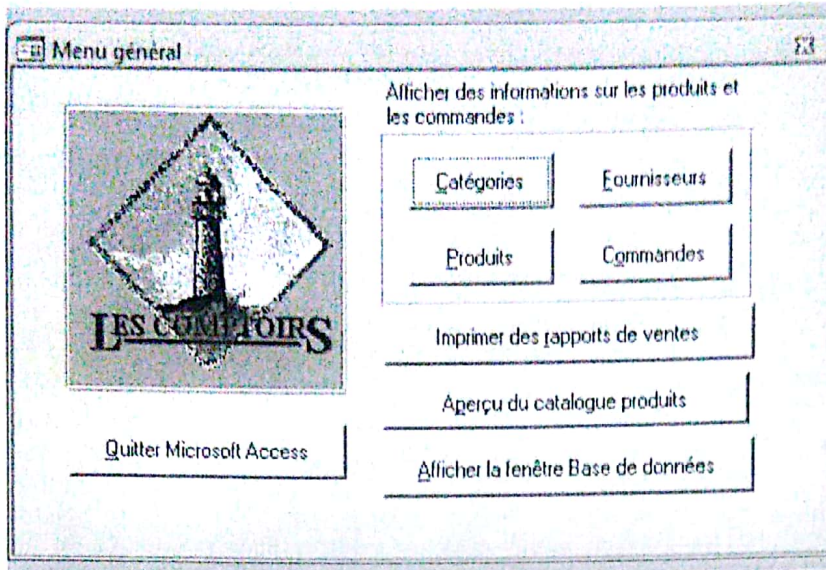
- ♣ Développer une interface avec Access c'est construire des formulaires avec ou sans macros
- ♣ Les **formulaires** sont des objets de la BD Access qui permettent
  - ◆ de visualiser des données de table(s) ,
  - ◆ d'ajouter/modifier/supprimer des données de(s) table(s) ou
  - ◆ d'afficher des résultats de requêtes à l'écran, ou
  - ◆ de créer des boites de dialogue ou
  - ◆ de créer des menus
- ♣ Les **formulaires** sont composés de **contrôles** (ou objets graphiques) dont l'objectif principal est d'afficher des données
- ♣ On peut créer des formulaires avec ou sans macro

©V. Heiwy

9

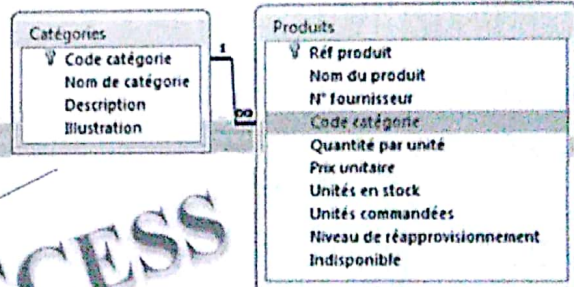
# Exemple de formulaire « menu »

ACCESS



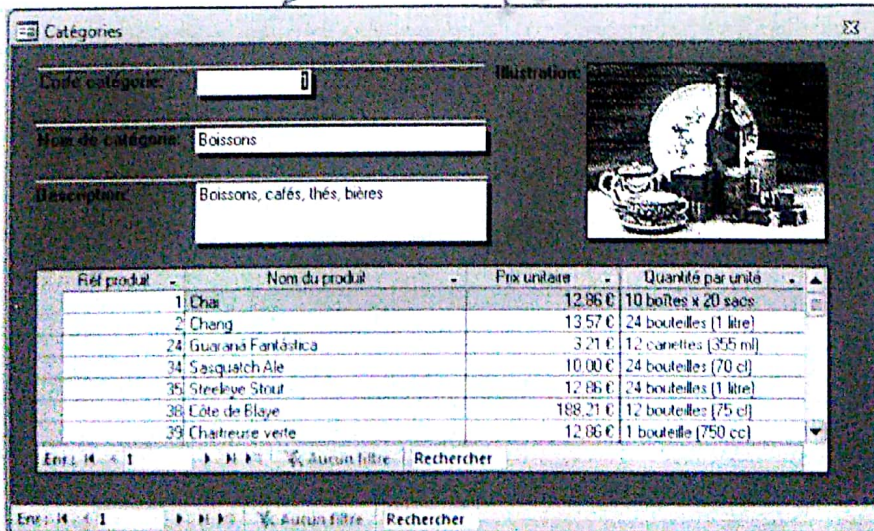
10

# Exemple de formulaire avec sous formulaire



Formulaire Catégories

ACCESS



Sous-Formulaire Produits

11

# Exemple de formulaire avec macro

ACCESS

Commandes

Facturé à: Alfreds Futterkiste  
 Obere Str. 57  
 Berlin 12209  
 Allemagne

Envoyé à: Alfreds Futterkiste  
 Obere Str. 57  
 Berlin 12209  
 Allemagne

Représentant: Peacock, Margaret  
 Messenger:  Speedy  United  Federal

N° comm.: 10702 Date comm.: 13/10/1997 Livrer avant: 24/11/1997 Date envoi: 21/10/1997

Produit	Prix unitaire	Quantité	Premise (%)	Prix total
Lakkalkooni	12,86 €	15	0%	192,86 €
Aniseed Syrup	7,14 €	6	0%	42,86 €
Sous total:				235,72 €
Port:				119,70 €
Total:				355,42 €

Enr: 14 1 sur 830 Rechercher

© V. Heivry

## Propriétés d'un formulaire

Clic droit ici

Type de sélection: Formulaire

Formulaire

Format	Données	Événement	Autres	Toutes
Source			SELECT C...	
Légende			Commandes	
Fen indépendante			Non	
Fen modale			Non	
Affichage par défaut			Formulaire unic	
Autoriser le mode Formulaire			Oui	
Autoriser le mode Feuille de			Oui	
Autoriser le mode Tableau cr			Oui	
Autoriser le mode Graphique			Oui	
Autoriser le mode Page			Non	
Type image			Intégré	
Image			(aucun)	
Mosaïque d'images			Non	
Alignement de l'image			Centre	
Mode d'affichage de l'image			Découpage	
Largeur			15,714cm	
Auto centrer			Oui	
Taille ajustée			Oui	
Ajuster à l'écran			Non	
Style bordure			Dimensionnable	
Afficher sélecteur			Non	
Boutons de déplacement			Oui	
Légende de la navigation			Non	
Diviseurs d'enregistrements			Oui	
Barre défilement			Aucun	
Boîte contrôle			Oui	
Bouton Fermer			Oui	
Boutons MinMax			Les deux actifs	
Déplaçable			Oui	
Taille du formulaire double x			Auto	
Orientation du formulaire de			Feuille de don	
Barre Fractionnement du for			Oui	
Feuille de données du formu			Modé autorisé	
Impression du formulaire des			Formulaire unic	
Enregistrer le positionnemen			Oui	
Sous-feuille données étendu			Non	
Sous-feuille données hauteur			0cm	
Grille X			16	
Grille Y			16	
Poices imprimante			Non	
Orientation			De gauche à d	
Type Recordset			Feuille de répo	
Extraction des paramètres pa			Oui	
Filtre				
Filtre sur chargement			Non	
Tri par				
Trier par sur chargement			Oui	
Attendre post-traitement			Non	

ACCESS

© V. Heivry

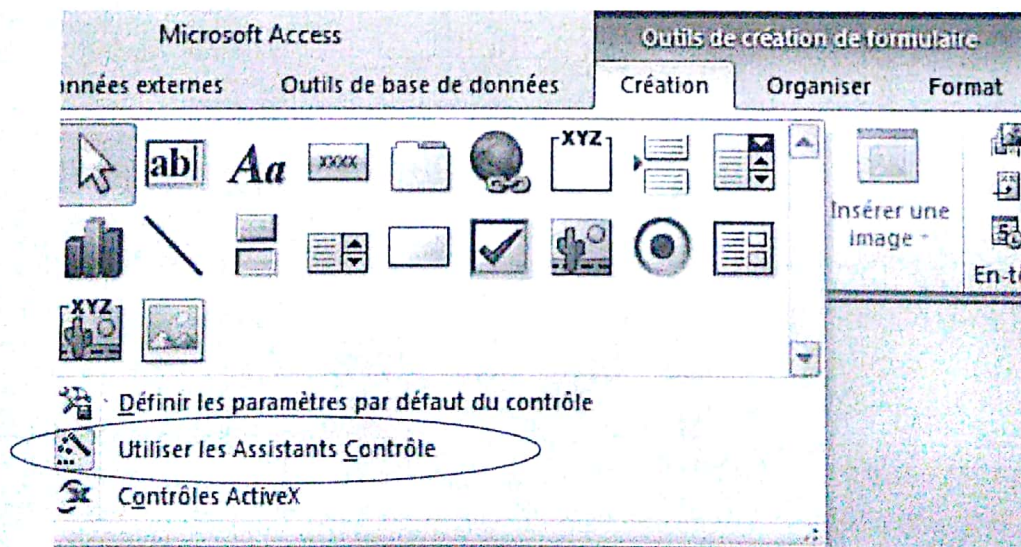
## Développement d'interface avec ACCESS

- ❖ Un contrôle est un objet graphique. On le retrouve dans les **formulaires** ou dans les **états**.
- ❖ Ils peuvent être **indépendants** des données de la base, ou **liés** aux données de la base ou **calculés**
- ❖ Il existe différents types de contrôles:
  - ◆ Zone de texte
  - ◆ Etiquette/ libellé
  - ◆ Liste de choix
  - ◆ Case à cocher
  - ◆ Bouton de commande ...
- ❖ Chaque contrôle est caractérisé par des **propriétés** précisant leur **nom**, la **source** de leur données et les **événements** qui s'y rattachent

14

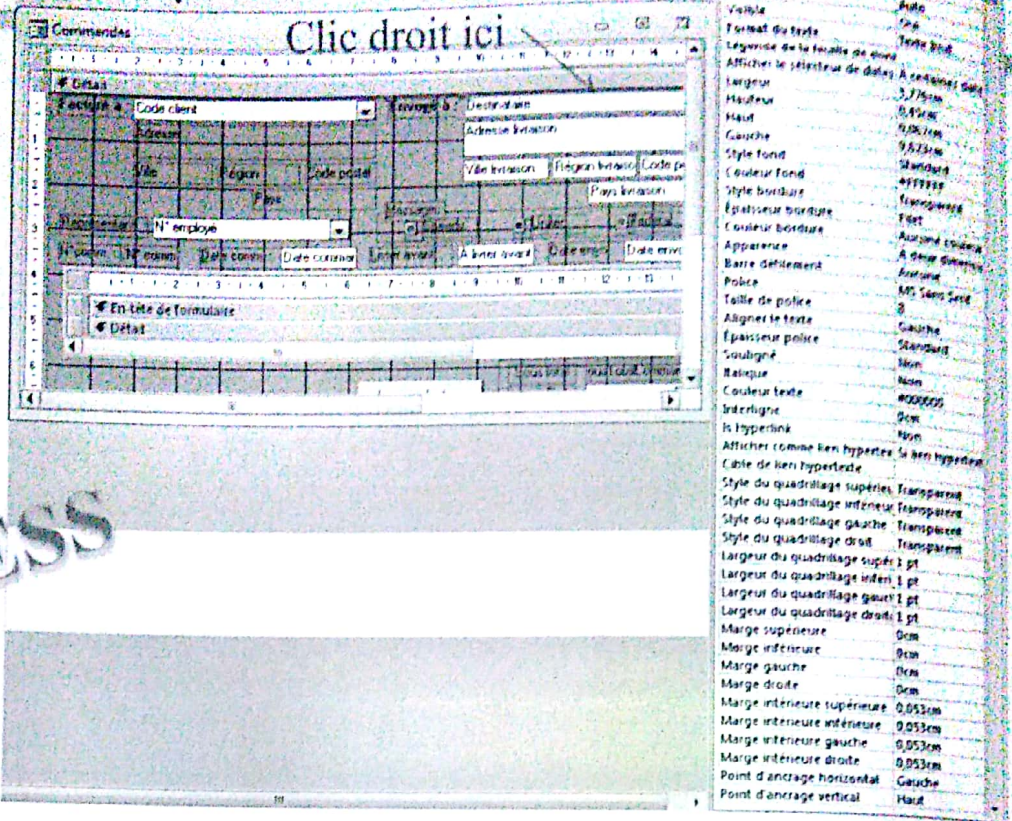
## Liste des contrôles

ACCESS



15

# Feuille de propriété d'un contrôle

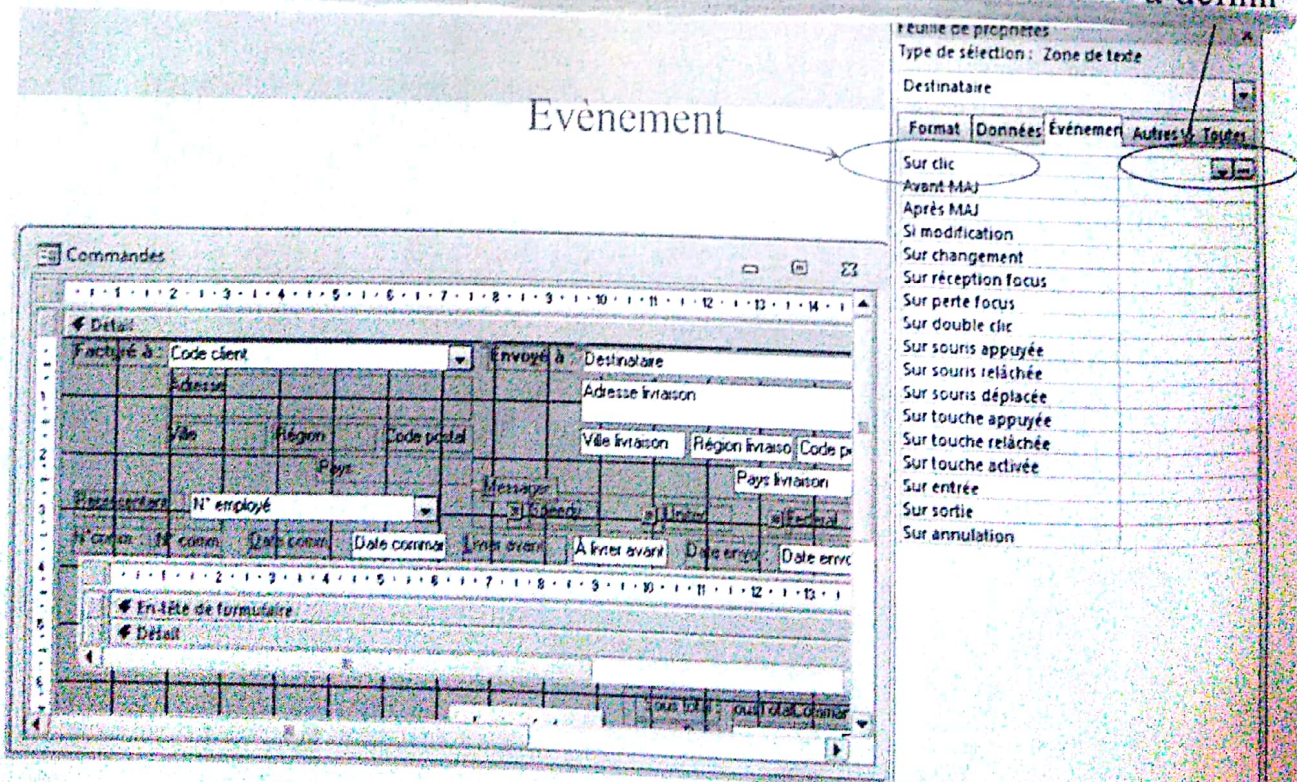


ACCESS

© V. Heivy

# Les propriétés « événements » d'un contrôle

ACCESS Macro à définir



© V. Heivy

# Développement d'interface avec ACCESS

✦ Les macros sont des suites d'actions qui s'exécutent suite à la survenance d'un évènement

✦ Comment créer une macro?

- A partir du menu Créer d'Access, ou
- A partir de l'évènement qui le déclenche, par exemple l'évènement « sur clic » d'un bouton de commande
- Un assistant peut être activé pour faciliter la création de la macro.

18

## Exemple de macro ACCESS

**Macro attachée au formulaire Clients.**

**Macro attachée à l'évènement Après MAJ du champ Code client.**

Sous-macro : ValiderCode

Si Erreur([vale client];[Client]);([code client] < Form.[code client]) Est Pas Null Alors

**Si la valeur de Code client n'est pas unique, afficher un message.**

**ZoneMessage**

Message Le code client que vous avez tapé existe déjà. Tapez un numéro unique.

Boj Oui

Type Information

Obj Code client dupliqué

**Revenir au contrôle Code client.**

Annuler l'évènement

Fin Si

**Macro attachée à l'évènement Après MAJ du formulaire.**

Sous-macro Fin

Sous-macro : MAJlistePays

**Actualise le contrôle Pays.**

Actualiser

Nom du contrôle Pays

Sous-macro Fin

Ajouter une nouvelle action

**Catalogue d'événements**

Rechercher

Déroulement de programme

- Commandes
- Groupe
- Li
- Tout Macro

Actions

- Commandes d'interface utilisateur
- Commandes de macro
- Commandes système
- Filtre/Requête/Recherche
- Gestion de fenêtre
- Importation/exportation de données
- Objets de base de données
- Opérations d'entrée de données

Dans cette base de données

# L'ERGONOMIE

© V. Heintz

20

## Qu'est-ce que l'ergonomie et plus spécifiquement l'ergonomie logicielle ?

- ♣ Le mot « ergonomie » vient du grec ergon (travail) et nomos (lois, règles).
- ♣ L'ergonomie peut donc être définie comme la science du travail ayant pour objet l'adaptation du travail à l'homme (amélioration des conditions de travail).
- ♣ Avoir comme objectif une meilleure adaptation du travail à l'homme implique (dans la mesure du possible) de considérer tous les aspects du travail : physiologiques, psychologiques, facteurs sociaux, facteurs objectifs et subjectifs.
- ♣ Une meilleure adaptation du travail à l'homme aura pour résultat la satisfaction des opérateurs, leur confort, leur santé mais aussi l'efficacité de leurs conduites opératoires.

© V. Heintz

source: <http://www.dsi.cnrs.fr/methodes/ergonomie/definition.htm>

21

## 2 types de problématiques en ergonomie

- ♣ L'ergonomie s'intéresse à deux grands types de problématiques :
  - ♦ problématique de l'adaptation de l'outil aux caractéristiques physiologiques et morphologiques de l'être humain ou d'une certaine population, c'est l'*ergonomie physique* ;
  - ♦ problématique de l'adaptation des outils au fonctionnement cognitif des utilisateurs, c'est l'*ergonomie cognitive*.

source: <http://www.dsi.cnrs.fr/methodes/ergonomie/definition.htm>

22

## L'ergonomie logicielle

- ♣ L'ergonomie logicielle se situe du côté de l'ergonomie cognitive.
- ♣ Son objectif est d'améliorer l'interaction homme - ordinateur ; faire en sorte que toute application informatique livrée aux utilisateurs soit :
  - ♦ utile : l'outil réalisé doit répondre aux besoins des utilisateurs pour lesquels il a été conçu, autrement dit en adéquation avec leur tâche
  - ♦ utilisable (ou maniable) : autrement dit facile à utiliser. L'utilisabilité recouvre trois choses :
    - ♥ la performance de réalisation de la tâche,
    - ♥ la satisfaction que procure l'utilisation de l'objet,
    - ♥ la facilité d'apprentissage.
- ♣ Dire qu'un logiciel est « ergonomique » ne veut rien dire en soi : un logiciel est ergonomique pour une personne donnée ayant un objectif donné.

source: <http://www.dsi.cnrs.fr/methodes/ergonomie/definition.htm>

23

## 2 niveaux d'ergonomie pour une application informatique (1/2)

- ♣ Un premier niveau que l'on pourrait appeler "l'ergonomie de surface" de l'application.
- ♣ Pour utiliser une métaphore, on pourrait dire que l'ergonomie de surface constitue la partie visible d'un iceberg, l'iceberg étant ici l'application.
- ♣ C'est ce que voit l'utilisateur de l'interface :
  - ♦ structuration des éléments dans les fenêtres/pages (disposition de ces éléments, densité d'affichage,...),
  - ♦ aspects graphiques (objets graphiques tels que boutons, logos, etc. et ligne graphique),
  - ♦ typographie,
  - ♦ réactions du système,
  - ♦ etc.

©V. Heiry

source: <http://www.dsi.cnrs.fr/methodes/ergonomie/definition.htm>

24

## 2 niveaux d'ergonomie pour une application informatique (2/2)

- ♣ Un second niveau que l'on pourrait appeler "l'ergonomie profonde" ; pour reprendre notre métaphore, l'ergonomie profonde constitue la partie immergée de l'iceberg, c'est-à-dire :
  - ♦ les fonctionnalités offertes à l'utilisateur ;
  - ♦ la structure de l'outil, autrement dit la répartition des fonctionnalités en modules, menus... ;
  - ♦ les enchaînements d'écran qui traduisent la succession des tâches et sous tâches que l'utilisateur a à effectuer.

©V. Heiry

source: <http://www.dsi.cnrs.fr/methodes/ergonomie/definition.htm>

25

# QUELQUES PRINCIPES D'ERGONOMIE POUR LES APPLICATIONS INFORMATIQUES

Source: Interfaces graphiques ergonomiques, Conception et Modélisation; Jean-Bernard CRAMPES; Ed. Ellipses; 1997.

- ♣ **LA COHERENCE:** Une même concept doit toujours être utilisé de façon similaire dans un contexte d'utilisation identique.
- ♣ **LA CONCISION:** Elle doit permettre de limiter le nombre d'interventions de l'utilisateur afin d'éviter les erreurs (par exemple typographiques).
- ♣ **LE RETOUR D'INFORMATIONS:** Toute action de l'utilisateur doit amener un retour d'information rapide et pertinent afin de lui permettre d'analyser rapidement le nouvel état de l'application.
- ♣ **LA STRUCTURATION DES ACTIVITES:** L'application doit être décomposée suivant une hiérarchie de niveaux de complexité croissante afin de proposer à l'utilisateur des fonctions simples.
- ♣ **LA FLEXIBILITE:** Toute application doit pouvoir être facilement personnalisée par les utilisateurs de façon à l'adapter à leurs propres habitudes et perception des choses.
- ♣ **LA GESTION DES ERREURS:** L'utilisateur doit être orienté vers une méthode lui permettant de résoudre son problème au lieu d'afficher un message élémentaire.

©V. Heivry

## COMMENT ASSURER LA COHERENCE?

Source: Interfaces graphiques ergonomiques, Conception et Modélisation; Jean-Bernard CRAMPES; Ed. Ellipses; 1997.

### ♣ cohérence avec le matériel

- ♥ Adaptation à l'écran (définitions et palettes de couleur),
- ♥ au clavier (touches de fonctions, touches spéciales),
- ♥ à la souris (nombre de boutons utilisables),
- ♥ à la carte son (désactivation des messages sonores si aucune carte n'est détectée).

### ♣ cohérence avec le logiciel

- ♥ les modules nécessaires (runtime, dll) au bon fonctionnement de l'application doivent être présents sur les ordinateurs d'accueil.

### ♣ cohérence intra-applications (interne)

- ♥ toutes les fenêtres et boîtes de dialogue doivent respecter des règles communes de présentation et d'organisation. Les menus et raccourcis claviers doivent avoir la même signification tout au long de l'application.

### ♣ cohérence inter-applications (externe)

- ♥ respect des règles communes à toutes les applications concernant les présentations, les méthodes d'interactions avec des objets identiques et les réponses fournies aux mêmes actions de l'utilisateur.

©V. Heivry

## COMMENT RENDRE L'INTERFACE TRANSPARENTE? (intuitive, visuelle, concise, explicative et flexible)

Source: Interfaces graphiques ergonomiques. Conception et Modélisation; Jean-Bernard CRAMPES; Ed. Ellipses; 1997

### ◆ Intuitive

- ♥ Utilisation instinctive avec un temps d'apprentissage réduit,
- ♥ Doit refléter le travail habituel de l'utilisateur sans l'ordinateur,
- ♥ Utilisation judicieuse de la souris, des icônes, des légendes et commentaires.

### ◆ Visuelle

- ♥ Affichage de menus d'actions et de listes de données au lieu de champs de saisie.
- ♥ Pré-remplir le champs de saisie avec une valeur qui servira d'exemple de référence.

### ◆ Concise

- ♥ Utiliser des abréviations pour limiter la saisie,
- ♥ Utiliser des raccourcis clavier et icônes pour accélérer l'exécution des commandes,
- ♥ Utiliser des macro-commandes pour personnaliser l'interface et définir plusieurs niveaux d'abstraction,
- ♥ Utiliser les copier-coller et les valeurs par défaut pour réutiliser les informations déjà saisies.

28

## COMMENT RENDRE L'INTERFACE TRANSPARENTE? (intuitive, visuelle, concise, explicative et flexible)

Source: Interfaces graphiques ergonomiques, Conception et Modélisation; Jean-Bernard CRAMPES; Ed. Ellipses; 1997.

### ◆ Explicative

- ♥ Proposer une aide (Aide ou « ? ») accessible depuis la barre d'actions
- ♥ Proposer une aide contextuelle accessible à partir de boutons de boîte de dialogue ou de touches de fonctions (F1, ...).
- ♥ Orienter l'utilisateur vers une solution en cas d'erreur détectée.

### ◆ Flexible

- ♥ Doit s'adapter à la configuration matérielle et logicielle automatiquement.
- ♥ Doit pouvoir être personnalisée par l'utilisateur (création de raccourcis-clavier, ajout d'icônes dans les barres d'outils, modification de valeurs par défaut, réorganisation de menus)
- ♥ Plus l'application s'adresse à un grand nombre d'utilisateurs, plus il faut lui permettre d'être personnalisable.

29

# CONCEPTION D'INTERFACE GRAPHIQUE

©V. Heiwy

30

## EN QUOI CONSISTE LA CONCEPTION D 'INTERFACE GRAPHIQUE?

Source: Interfaces graphiques ergonomiques, Conception et Modélisation; Jean-Bernard CRAMPES; Ed. Ellipses; 1997.

- ♣ Définition des différentes fenêtres (primaires, secondaires, bôtes de dialogue, bôtes de messages),
- ♣ Organisation des fenêtres primaires et secondaires
  - ◆ 3 règles
- ♣ Organisation des boites de dialogue et des contrôles qu 'elles contiennent
  - ◆ 10 règles
- ♣ logique d 'enchaînement des fenêtres et des évènements
  - ◆ Diagramme d'états

©V. Heiwy

31

# CONCEPTION D 'INTERFACE GRAPHIQUE:

## (1) Définition des différentes fenêtres

Source: Interfaces graphiques ergonomiques, Conception et Modélisation; Jean-Bernard CRAMPES; Ed. Ellipses; 1997.

- ♣ En liaison avec un MOT MERISE ou un DCU UML
- ♣ En respectant les principes d 'ergonomie
  - ♦ énoncés précédemment
- ♣ **Fonctionnement des fenêtres**
  - ♦ Diagramme d 'états pour gérer les interactions ayant lieu à l'intérieur d'une fenêtre (réalisation des contrôles, affichage des messages d'erreur).
- ♣ **Diagramme d 'états pour représenter les enchaînements entre les fenêtres**

©V. Heivy

32

# CONCEPTION D 'INTERFACE GRAPHIQUE:

## (2) Organisation des fenêtres primaires et secondaires

Source: Interfaces graphiques ergonomiques, Conception et Modélisation; Jean-Bernard CRAMPES; Ed. Ellipses; 1997.

- ♣ **Règle 1: Taille de la barre d 'actions** (elle ne doit occuper qu'une seule ligne dans la fenêtre)
- ♣ **Règle 2: Taille des menus** (la barre des menus doit contenir 12 options au maximum et la barre d'actions pas plus de trois niveaux)
- ♣ **Règle 3: Existence de la région client** (si elle n'est pas utilisée, elle doit être supprimée)

## CONCEPTION D 'INTERFACE GRAPHIQUE:

### (3) Organisation des bôîtes de dialogue et des contrôles

Source: Interfaces graphiques ergonomiques, Conception et Modélisation; Jean-Bernard CRAMPES; Ed. Ellipses; 1997.

- ◆ **Règle 4 : Nombre de contrôles (<30)**
- ◆ **Règle 5: Taille et position des boîtes de dialogue** (minimum de surface et proche de l'objet qui l'a appelé sans le recouvrir)
- ◆ **Règle 6: Position et regroupement des contrôles** (par famille avec cadre de groupage, alignement des cadres et titres pour les contrôles sans légende)
- ◆ **Règle 7: Mise en Majuscule** (la 1ere lettre de chaque chaîne de caractères sauf pour les sigles et abréviations)
- ◆ **Règle 8: Valeurs initiales** (donner une valeur plausible chaque fois que c'est possible)

## CONCEPTION D 'INTERFACE GRAPHIQUE:

### (3) Organisation des bôîtes de dialogue et des contrôles

Source: Interfaces graphiques ergonomiques, Conception et Modélisation; Jean-Bernard CRAMPES; Ed. Ellipses; 1997.

- ◆ **Règle 9: Types de Listbox** (listbox ou combo-box simples plutôt que déroulantes, visualisation de 3 éléments au moins sans scrolling)
- ◆ **Règle 10: Choix et disposition des boutons poussoirs** (au même endroit dans toutes les boites de dialogue et dans le même ordre)
- ◆ **Règle 11: Taille et structuration des champs de saisie** (adaptée au type et à la longueur des données à saisir)
- ◆ **Règle 12: Couleurs et polices** (homogénéité dans toute l'application et avec les autres applications réalisées; par exemple, utiliser les couleurs pour distinguer les champs à saisir des champs affichés par l'application)
- ◆ **Règle 13: Aide en ligne** (prévoir les fonctionnalités générales de l'application et le fonctionnement de chaque boite de dialogue).

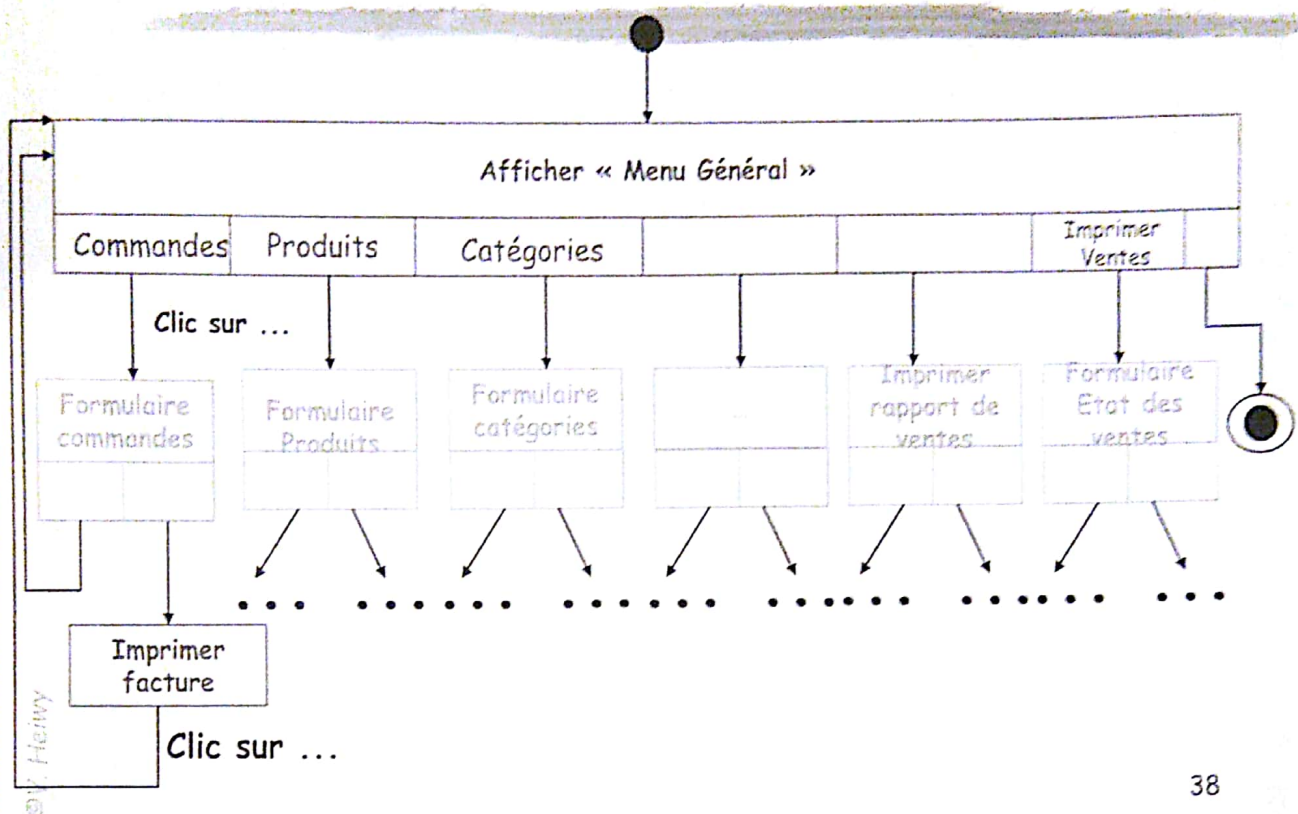
## Conception d'une interface graphique: (4.1) logique d'enchaînement des fenêtres

- ✿ Pour représenter les enchaînements, il faut prendre en compte
  - ◆ Les événements (clic de souris, passage au dessus d'un objet, etc.
  - ◆ L'événement ouverture de la fenêtre: du code executant des instructions peut y être associé
  - ◆ Le découpage en modules (dans le cas d'applications volumineuses)
  - ◆ Les paramètres transmis (lors de l'ouverture d'une fenêtre)

## CONCEPTION D'INTERFACE GRAPHIQUE: (4.2) modélisation de l'enchaînement des fenêtres

- ✿ Utilisation de scénarios pour décrire tous les cas d'utilisation de l'interface
  - ◆ scénario normal
  - ◆ scénario d'erreur
  - ◆ scénario d'exception
- ✿ Utilisation d'un diagramme d'états pour décrire l'enchaînement des fenêtres, des boîtes de dialogue et des boîtes de messages
  - ◆ Les états représentent l'affichage d'une fenêtre active et le travail de l'utilisateur sur cette fenêtre. L'état se termine par l'appui sur une touche ou sur l'activation d'un bouton « fin de fenêtre ».
  - ◆ La transition prend en compte les données de la fenêtre correspondant à l'ETAT déclenchant la transition, le traitement en fonction des activations fait apparaître une autre fenêtre.

EXEMPLE "LES COMPTOIRS":  
GENERER UN DIAGRAMME D'ETATS POUR L'ENCHAINEMENT DES FENETRES

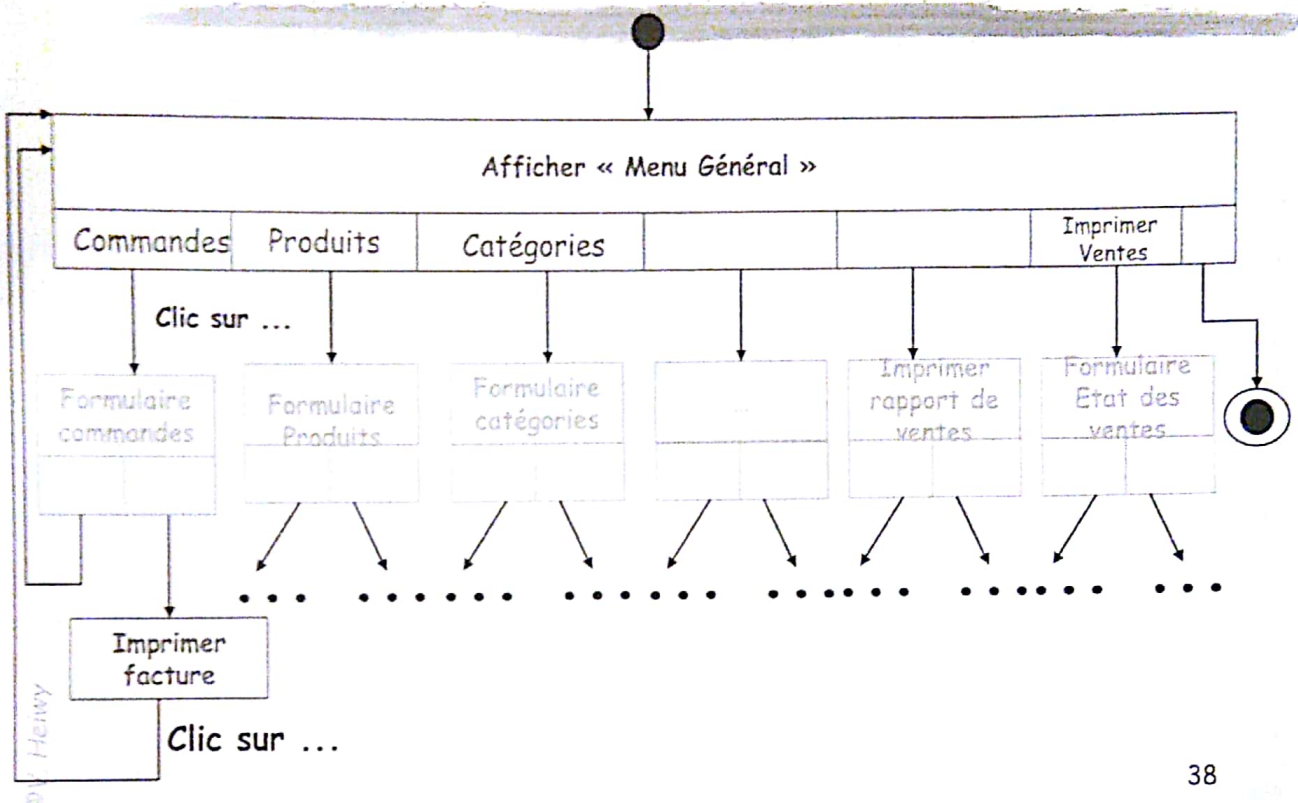


38

# MAQUETTAGE ET ENCHAINEMENT DE FENETRES

39

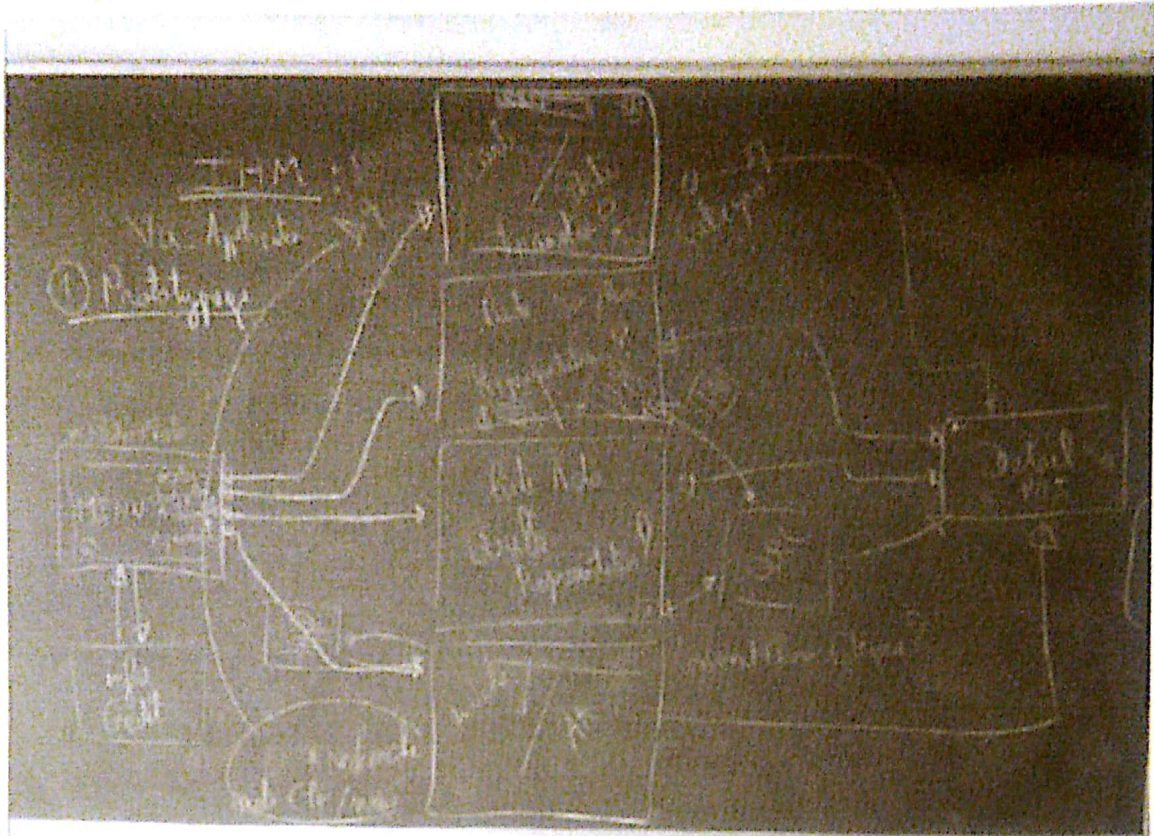
EXEMPLE "LES COMPTOIRS":  
GENERER UN DIAGRAMME D'ETATS POUR L'ENCHAINEMENT DES FENETRES



# MAQUETTAGE ET ENCHAINEMENT DE FENETRES

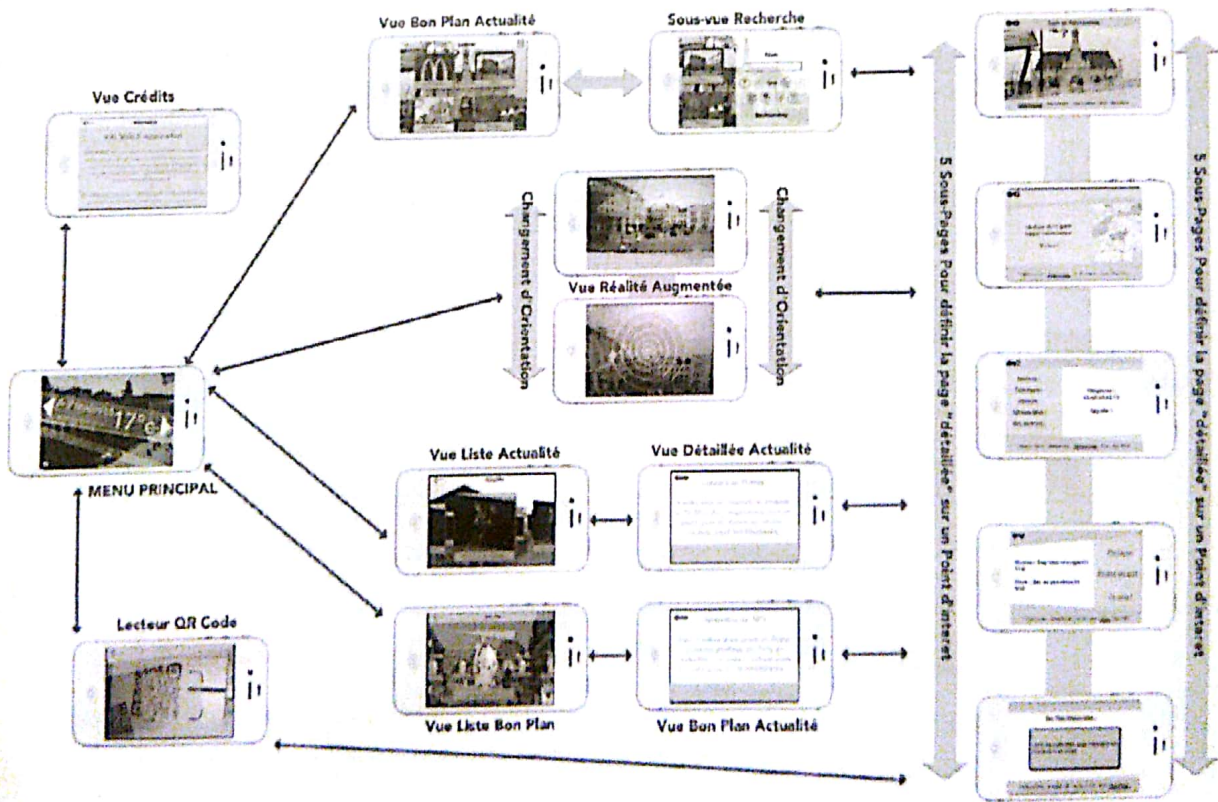


# Maquettage tableau



Première maquette d'une application mobile sur iPhone et iPad , avec réalité augmentée  
(N. LAURENT et G.KOSSI, Projet M2 TNSI, 2013)

© C. Kolski



Prototypage d'une application mobile, version iPhone, avec réalité augmentée  
(N. LAURENT et G.KOSSI, Projet M2 TNSI, 2013)

© C. Kolski

## Maquettage sur tableau blanc

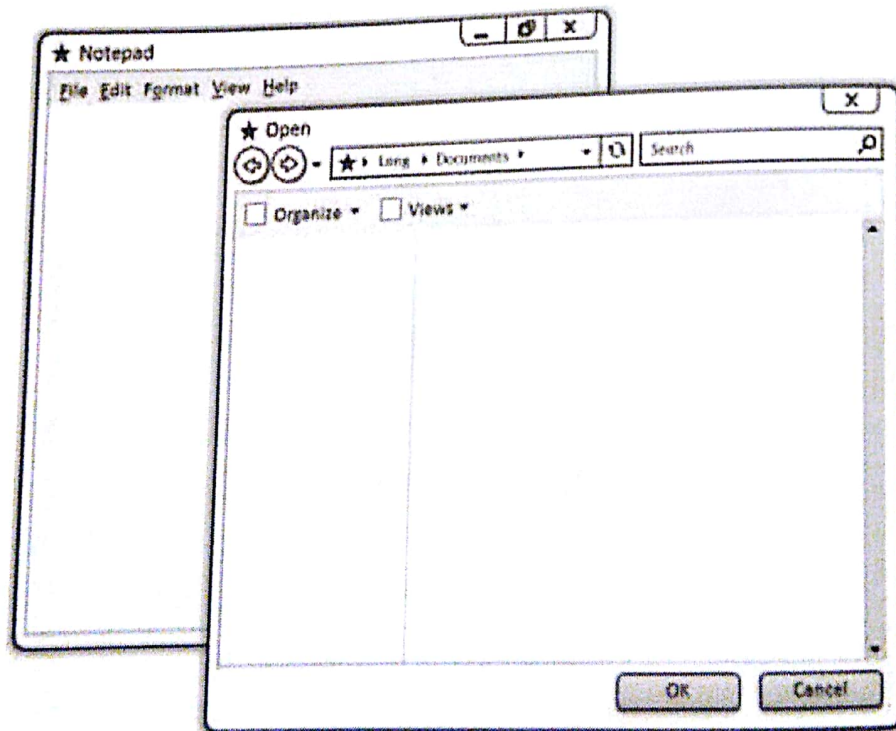


“Video Prototyping: The CPN design team reviews their observations of CPN developers and then discuss several design alternatives. They work out a scenario and storyboard it, then shoot a video prototype that reflects their design.”  
(BEAUDOUIN-LAFON et MACKAY, 2002)

*Rem : ne pas oublier l'appareil photo !*

© C. Kolski

## Maquettage avec un éditeur de présentation



Consulter :

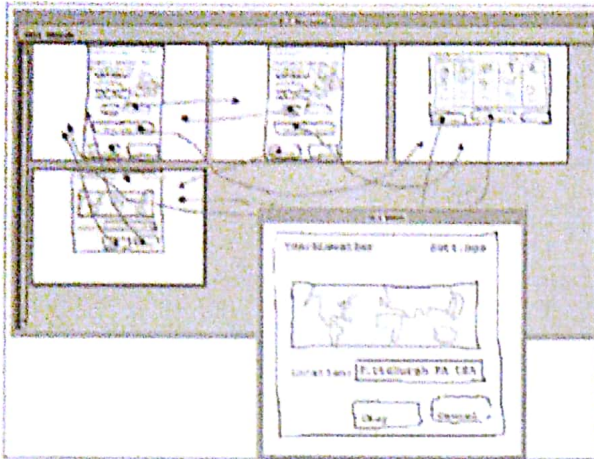
<http://www.fredcavazza.net/2007/10/24/des-gabarits-de-creation-de-prototypes-avec-powerpoint/>  
(de même que ses blogs et présentations...)

Nous utiliserons l'outil de maquettage OpenElement pour le projet du 2<sup>ème</sup> semestre

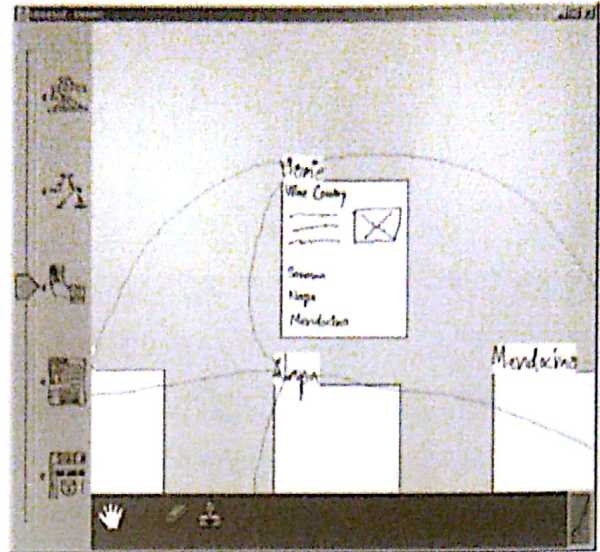
© C. Kolski

# Maquettage & prototypage, quelques modèles et méthodes

- Editeurs orientés maquettage, exemples :



SILK (LANDAY, PhD 1996) :  
fidélité basse, présentationnel



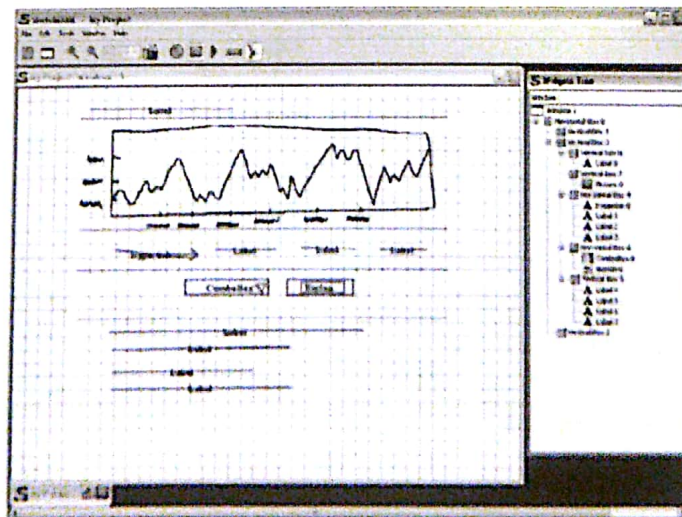
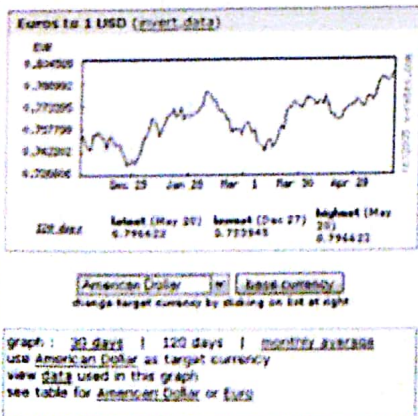
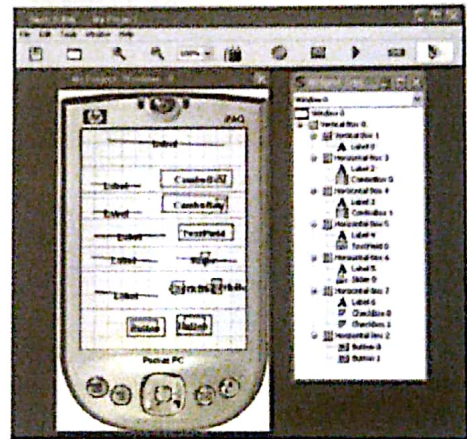
DENIM (LIN et al., 2000) :  
fidélité basse, présentationnel et  
navigationnel

© C. Kolaki

# Maquettage & prototypage, quelques modèles et méthodes

-Travaux autour de UsiXML  
(User Interface exTensible Markup Language),  
<http://www.w3.org/2005/Incubator/model-based-ui/wiki/UsiXML>

Fidélité basse: SketchiXML  
Fidélité moyenne: VisiXML  
Fidélité élevée: GrafiXML



© C. Kolaki (voir aussi : VANDERDONCKT et COYETTE, 2007)

# Maquettage du contexte d'interaction

(BEAUDOUIN-LAFON et MACKAY, 2002)

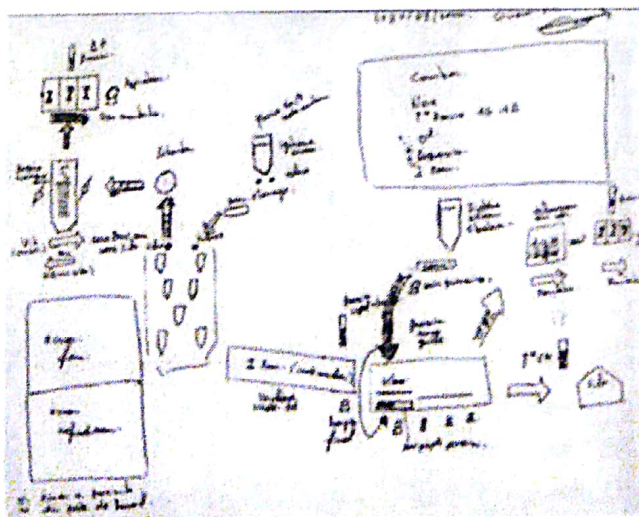
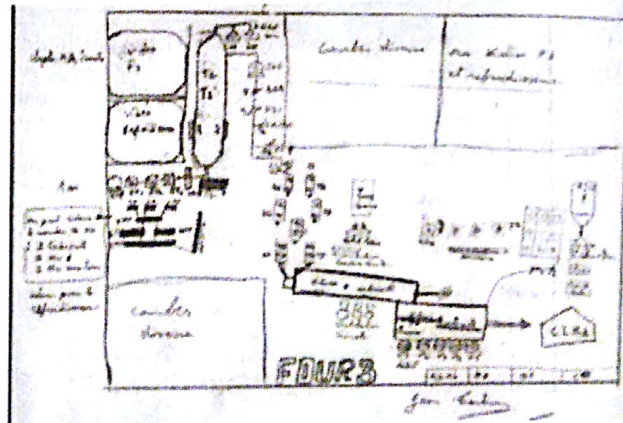


Scaled mock-up of an air traffic control table, connected to a wall display

© C. Kolski

## Maquettage des IHM par leurs propres utilisateurs

Cas d'IHM de supervision  
d'une cimenterie :  
(SAGAR, 2003)



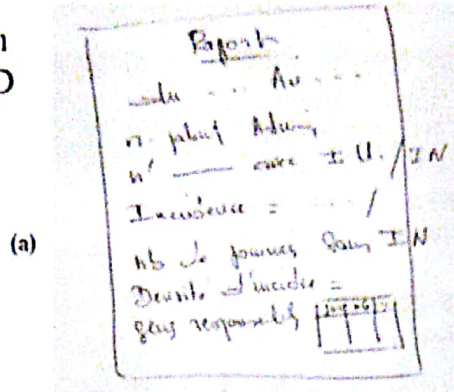
Deux exemples parmi les  
maquettes réalisées par les  
opérateurs humains

© C. Kolski

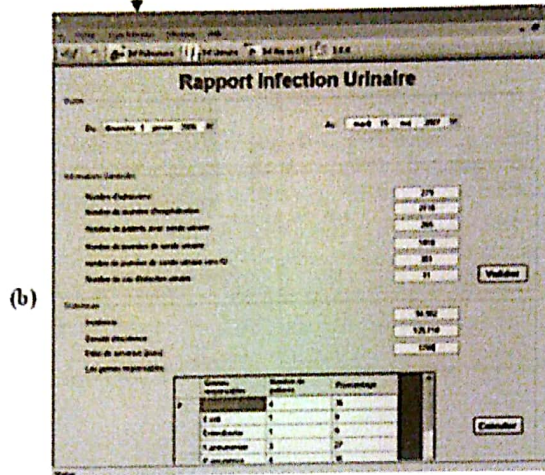
(Voir aussi KOLSKI et WAROUX (1992) dans le domaine gazier)

# Maquettage des IHM par leurs propres utilisateurs

- De l'expression du besoin par le médecin à un SIAD pour la lutte contre les infections nosocomiales, selon une démarche itérative



Progressive modifications based on user remarks and the results of the user-centered tests

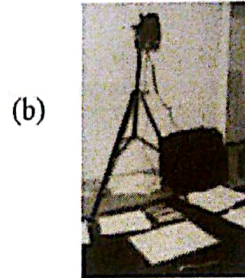
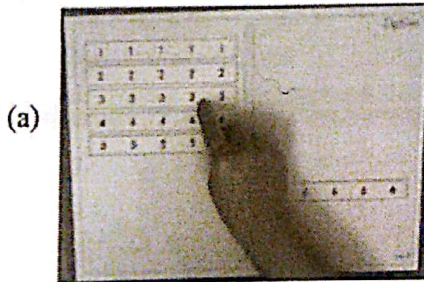


(BEN AYED et al., 2010)

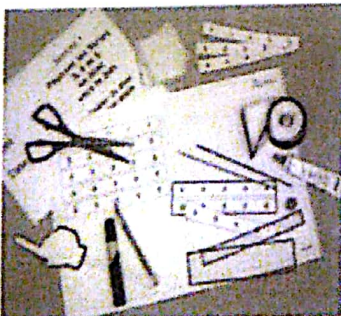
© C. Kolski

## Maquettage video

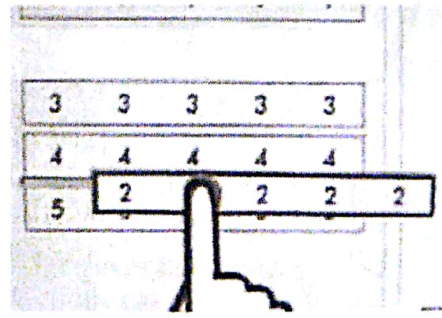
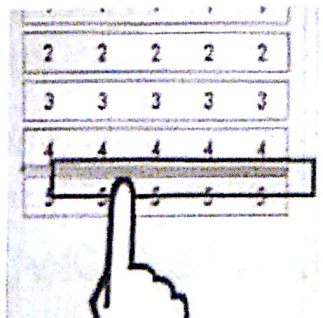
(MERTZ et VINOT, 1998)



Un sujet au cours du maquettage papier (a) et la caméra au dessus de l'écran « tactile » (b)



Matériel utilisé



Extraits d'animations :  
déplacement du strip2 avec fantôme avec drag-and-drop (c)

Dans ce cas : tournage image par image (1/4 ou 1/2 s)

© C. Kolski

## Spécification d'une application de gestion de trafic :

- Infrastructures routières : permettent de gérer le trafic autoroutier
- Utilisateurs : expert en sécurité, architecture, transports...
- Objets / agents tangibles :
  - Agents matériels (feux, panneaux de signalisation...)
  - Objets infrastructures
  - Tampon (création de nouveaux véhicules)
  - ...
- Objets / agents virtuels :
  - Routes
  - Véhicules
  - ...

## Application sur table interactive TangiSense, équipée de la RFID :



1<sup>ère</sup> version : maquettage video de l'objet tangible Zoom

2<sup>ème</sup> version : prototype (KUBICKI et al., 2013)

©C. Kolski

## UML: des diagrammes au service de la modélisation des interfaces

Modéliser l'enchaînement des fenêtres avec :

**UN DIAGRAMME D' ETATS**

**UN DIAGRAMME DE COLLABORATION**

Modéliser l'enchaînement des fenêtres:

## DIAGRAMME D' ETATS

© V. Henry

54

### PRESENTATION DU DIAGRAMME D' ETATS

- ♣ LE DIAGRAMME D' ETATS
- ♣ LES ETATS
- ♣ LES TRANSITIONS
- ♣ LES EVENEMENTS
- ♣ LES ACTIONS
- ♣ LES ACTIVITES
- ♣ EXEMPLES: APL et DAB



© V. Henry

55

# DIAGRAMME D'ETATS

- ♣ Le **DIAGRAMME D'ETATS** représente les traitements permettant de gérer le domaine étudié, les **TRAITEMENTS** (opérations) par rapport à des **ETATS** des classes (ici l'interface).
- ♣ Il met en évidence l'**enchaînement des états** et fait apparaître l'**ordonnancement** de ces différents travaux.
- ♣ Il permet de représenter tous les états possibles ainsi que les **EVENEMENTS** provoquant les changements d'état.

© V. Heiwy

56

## DIAGRAMME D'ETATS

### Les états

- ♣ Un **ETAT** est une situation durable dans laquelle peut se trouver l'interface (ou l'une de ses fenêtres) et à laquelle on associe des règles et des activités particulières.
- ♣ Les objets de certaines classes n'ont qu'un seul état pendant toute la durée de leur existence.
- ♣ L'état est représenté par un rectangle aux angles arrondis.

**Etat 1**  
activité *i*

- ♣ On passe d'un état à un autre suite à un événement.
- ♣ Tout événement ne provoque pas un changement d'état.

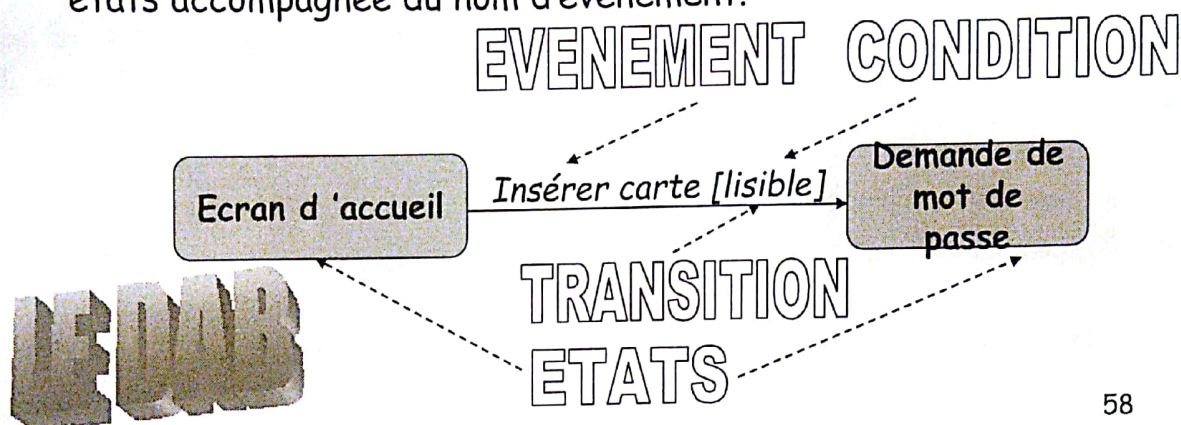
© V. Heiwy

57

# DIAGRAMME D'ETATS

## Les Transitions

- ♣ Une **TRANSITION** est une relation entre deux états signifiant qu'un passage de l'un à l'autre est possible.
- ♣ Elle est orientée.
- ♣ Un **EVENEMENT** y est attaché et indique qu'un objet passe d'un état à un autre sous certaines conditions.
- ♣ La transition est représentée par une flèche entre deux états accompagnée du nom d'événement.

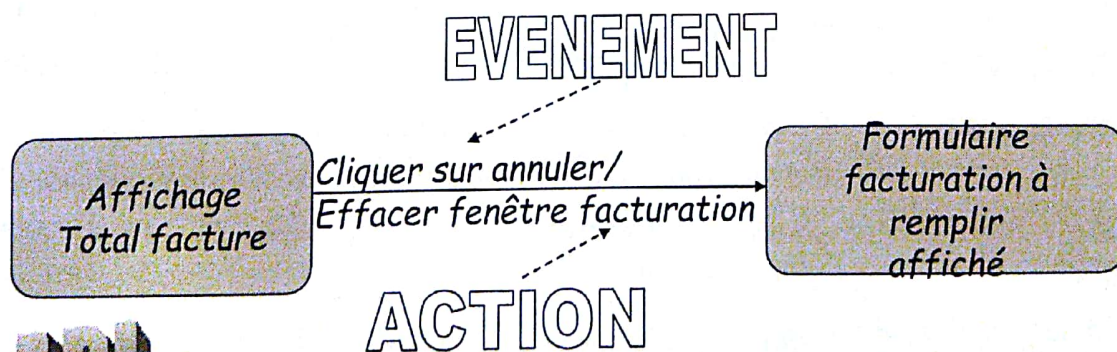


58

# DIAGRAMME D'ETATS

## Les actions

- ♣ Une **ACTION** est une opération élémentaire et instantanée.
- ♣ Elle est rattachée à la réalisation d'un événement, ou à l'entrée dans un état, ou à la sortie d'un état.
- ♣ Exemple: On associe à l'événement « cliquer sur annuler » l'action « Effacer fenêtre facturation ».

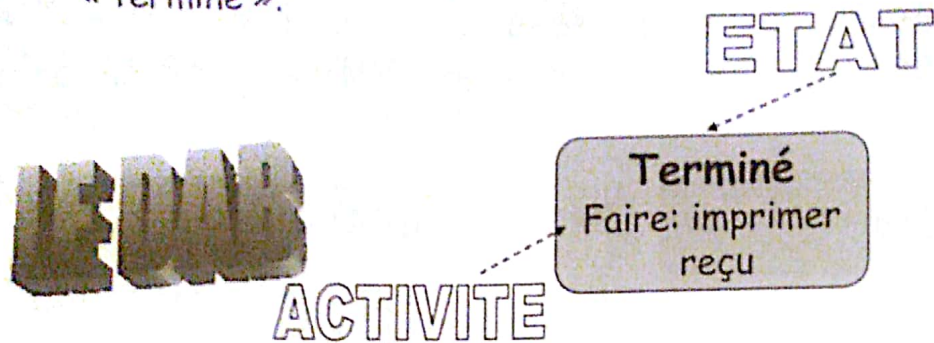


59

# DIAGRAMME D'ETATS

## Les activités

- ❖ Une **ACTIVITE** est une action qui dure.
- ❖ Chaque activité est associée obligatoirement à un état d'objet et vice-versa.
- ❖ **Exemple:** l'activité « imprimer reçu » est associée à l'état « terminé ».

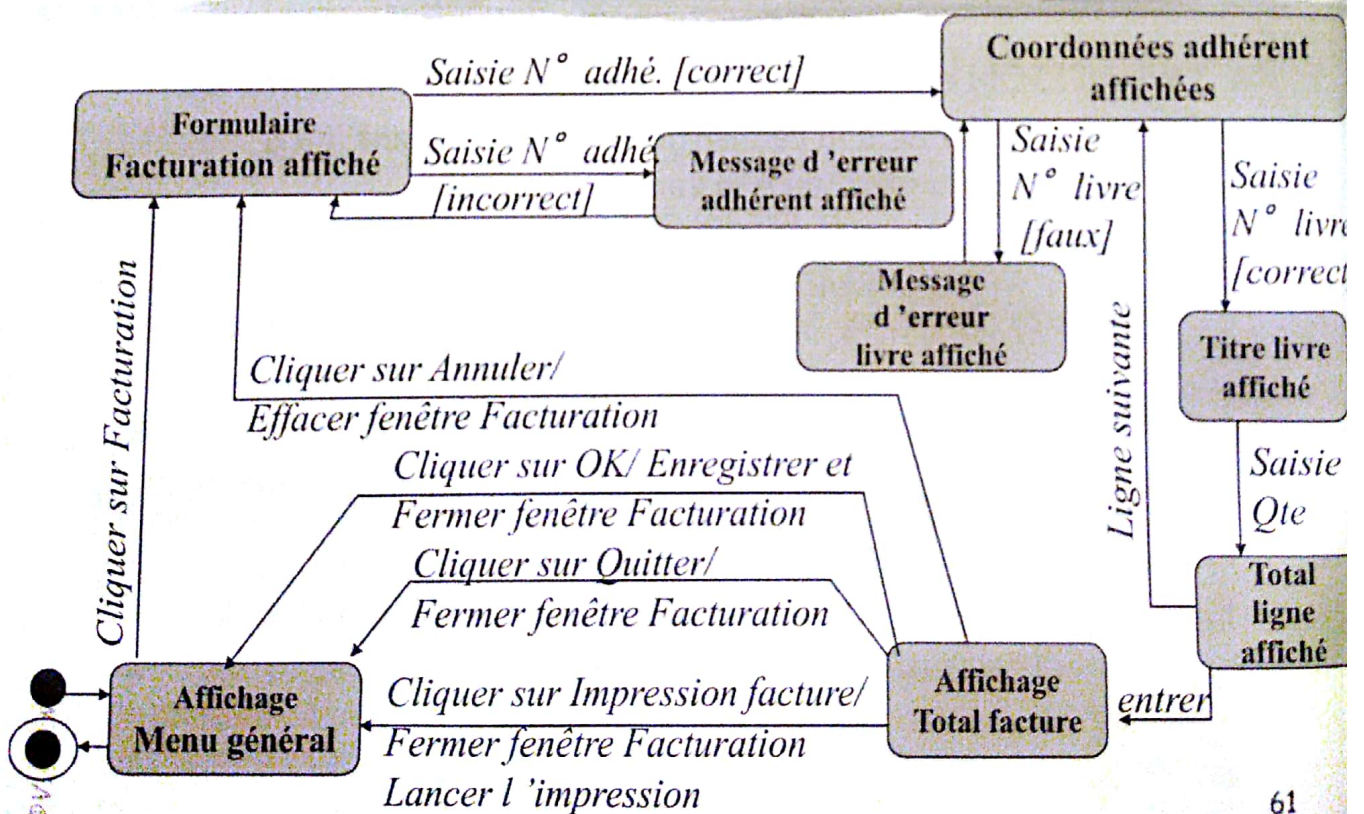


©V. Heuwy

60

# DIAGRAMME D'ETATS

## APL- La facturation



61

# SCENARIO

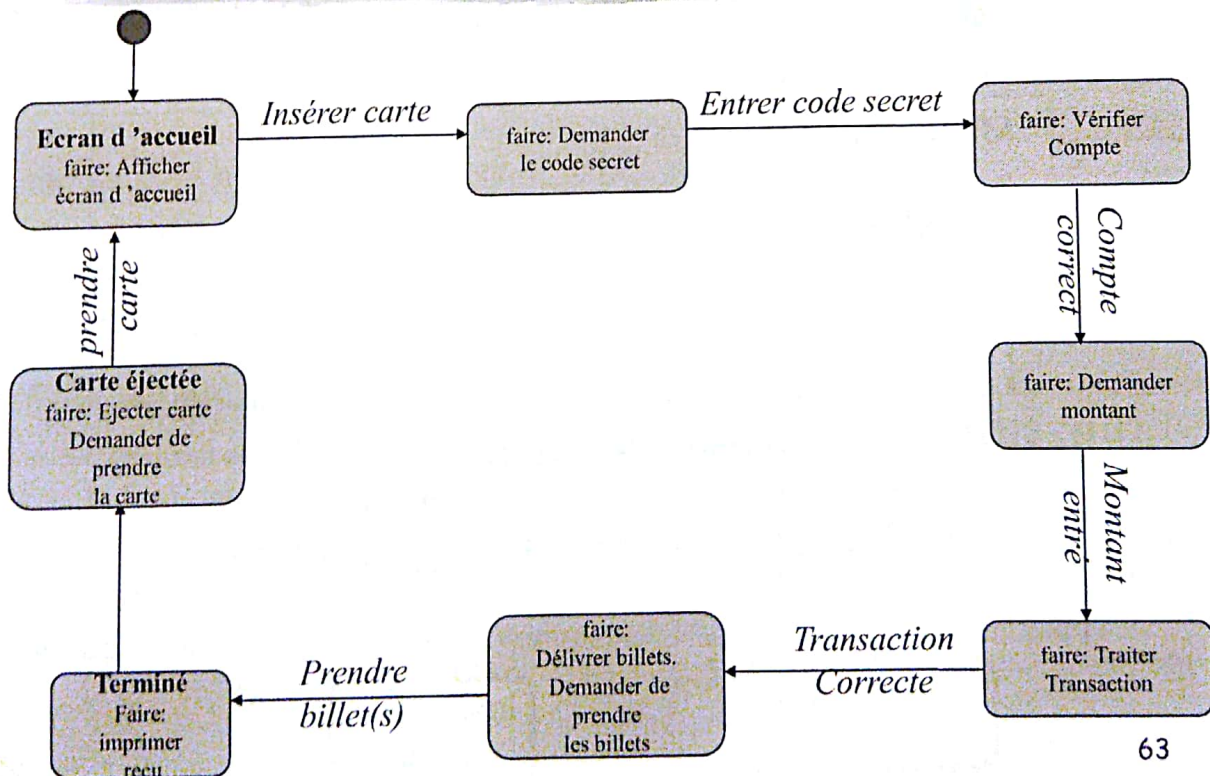
## le DAB - cas normal

- ♣ Le DAB demande à l'utilisateur d'insérer la carte. L'utilisateur insère la carte.
- ♣ Le DAB accepte la carte et lit le N° de série.
- ♣ Le DAB demande le code secret. L'utilisateur entre « 1.2.3.4 ».
- ♣ Le DAB vérifie le N° de série et le code secret par l'intermédiaire du consortium. Le consortium vérifie à l'aide de la banque « 39 » et signifie son acceptation au DAB.
- ♣ Le DAB demande à l'utilisateur de spécifier le montant de son retrait. L'utilisateur indique 500.
- ♣ Le DAB vérifie que le montant ne dépasse pas la somme autorisée et transmet le traitement de la transaction au consortium. Le consortium fait suivre vers la banque qui retourne éventuellement un acquittement avec le nouveau solde.
- ♣ Le DAB délivre l'argent et demande à l'utilisateur de le prendre. L'utilisateur le prend.
- ♣ Le DAB délivre le reçu, éjecte la carte et demande à l'utilisateur de les récupérer. L'utilisateur les récupère.

62

# DIAGRAMME D'ETATS

## le DAB - cas normal



63

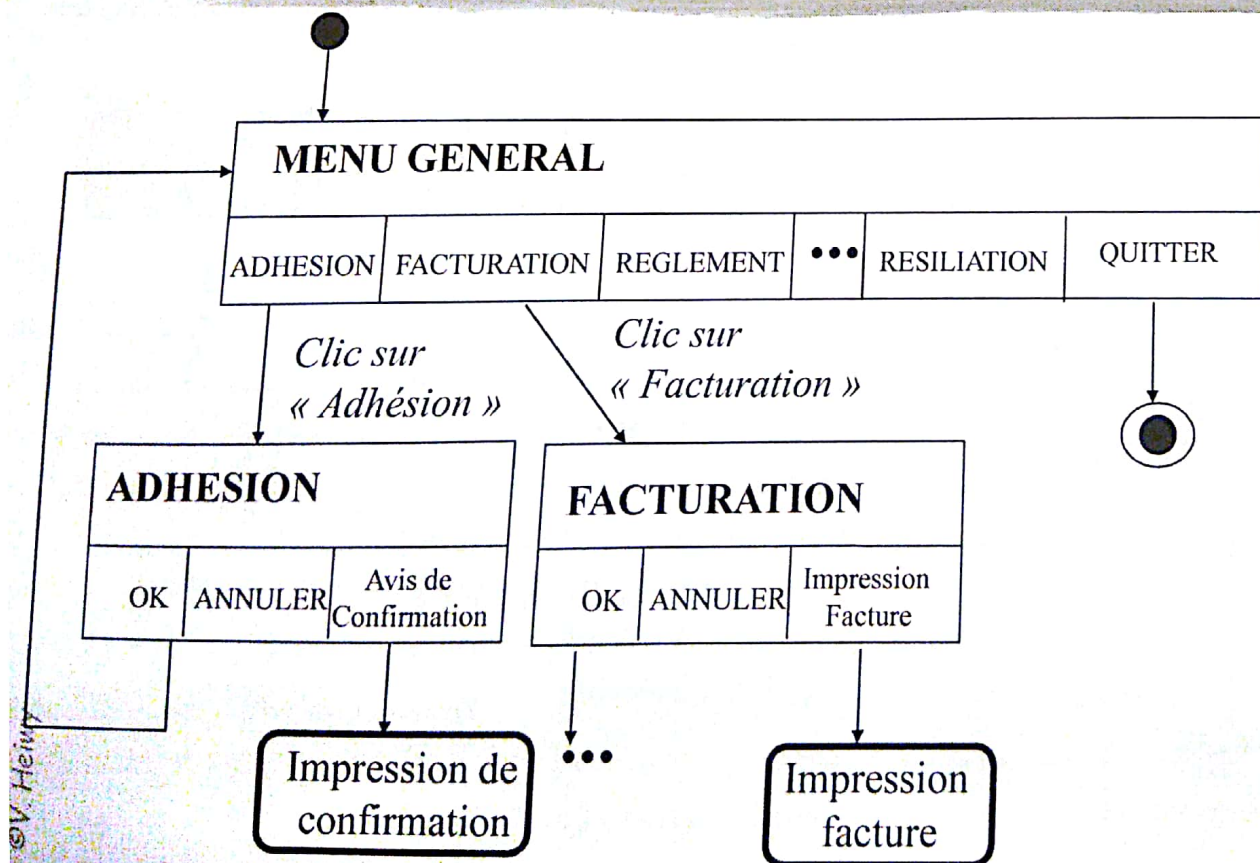
# MAQUETTE DE L'INTERFACE le DAB - cas normal



©V. Heivy

64

## CONCEPTION D'INTERFACE GRAPHIQUE: (4.2) Modélisation de l'enchaînement des fenêtres



©V. Heivy

65

# APPLICATIONS

## Création d'interface

### ♣ Le DAB

- ♦ Définir sous forme de scénario un cas d'erreur qui pourrait se présenter dans le déroulement d'un « Retrait d'argent ».
- ♦ Compléter le diagramme d'états pour prendre en compte ce nouveau scénario.
- ♦ Comment modifier le diagramme d'état représentant le fonctionnement du DAB pour prendre en compte le fait que l'on puisse réaliser d'autres opérations que des retraits (par exemple dépôt, virement, relevé)

66

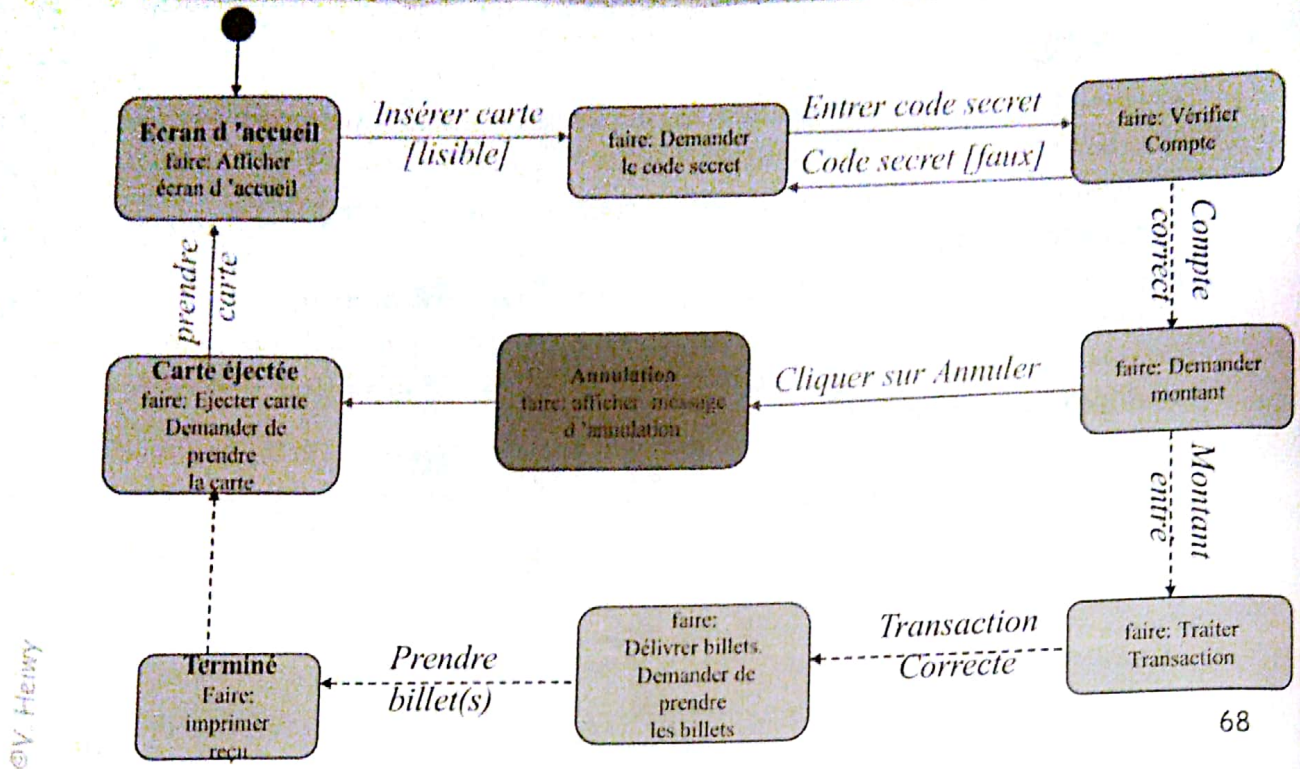
## SCENARIO

### le DAB - Un cas d'erreur

- ♣ Le DAB demande à l'utilisateur d'insérer la carte. L'utilisateur insère la carte.
- ♣ Le DAB accepte la carte et lit le N° de série.
- ♣ Le DAB demande le code secret. L'utilisateur entre « 9.9.9.9 ».
- ♣ Le DAB vérifie le N° de série et le code secret par l'intermédiaire du consortium. Le consortium rejette la carte après consultation de la banque appropriée.
- ♣ Le DAB indique qu'un mauvais code a été entré et demande à l'utilisateur de l'entrer à nouveau. L'utilisateur entre « 1.2.3.4 » que le DAB vérifie avec succès.
- ♣ Le DAB demande à l'utilisateur de spécifier le montant de son retrait. L'utilisateur change d'avis et appuie sur « Annuler ».
- ♣ Le DAB éjecte la carte et demande à l'utilisateur de la récupérer. L'utilisateur la retire.

67

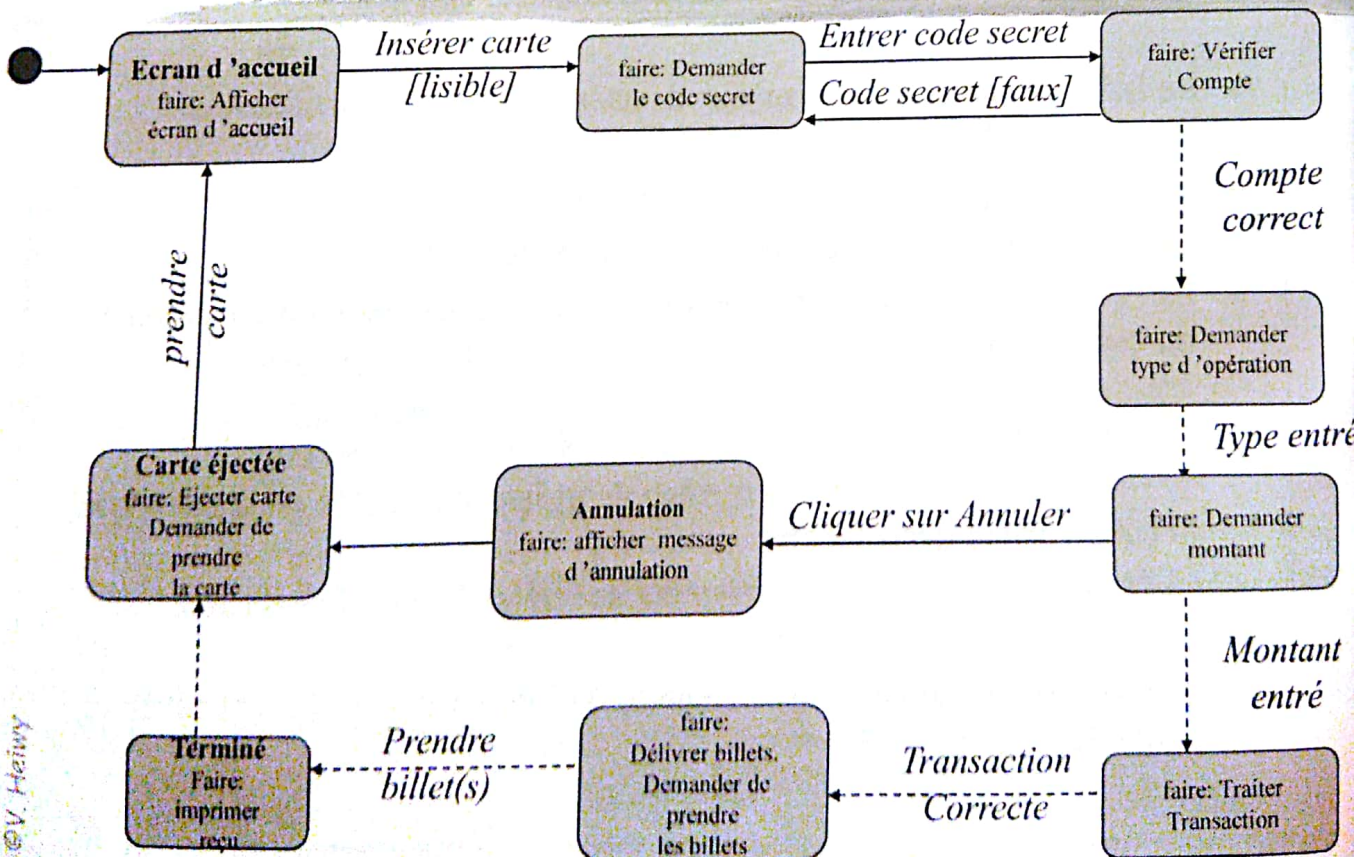
# DIAGRAMME D'ETATS le DAB - incluant le cas d'erreur



68

# DIAGRAMME D'ETATS le DAB - complété

Plan



# Conception/creation d'une interface graphique

---

- ♣ L'objectif de l'interface graphique est de rendre accessible à l'utilisateur les **fonctionnalités** (décrites par exemple sous forme de cas d'utilisation (DCU d'UML) ou sous forme de diagramme de collaboration d'UML (voir transparent suivant)) de l'application de la manière la plus ergonomique et simple possible
- ♣ Aspect "statique" de l'interface: réalisation des formulaires
- ♣ Aspect "dynamique" de l'interface:
  - ♣ décrire l'enchaînement des fenetres
  - ♣ Programmer les traitements automatisés

70

## PRESENTATION DU DIAGRAMME DE COLLABORATION

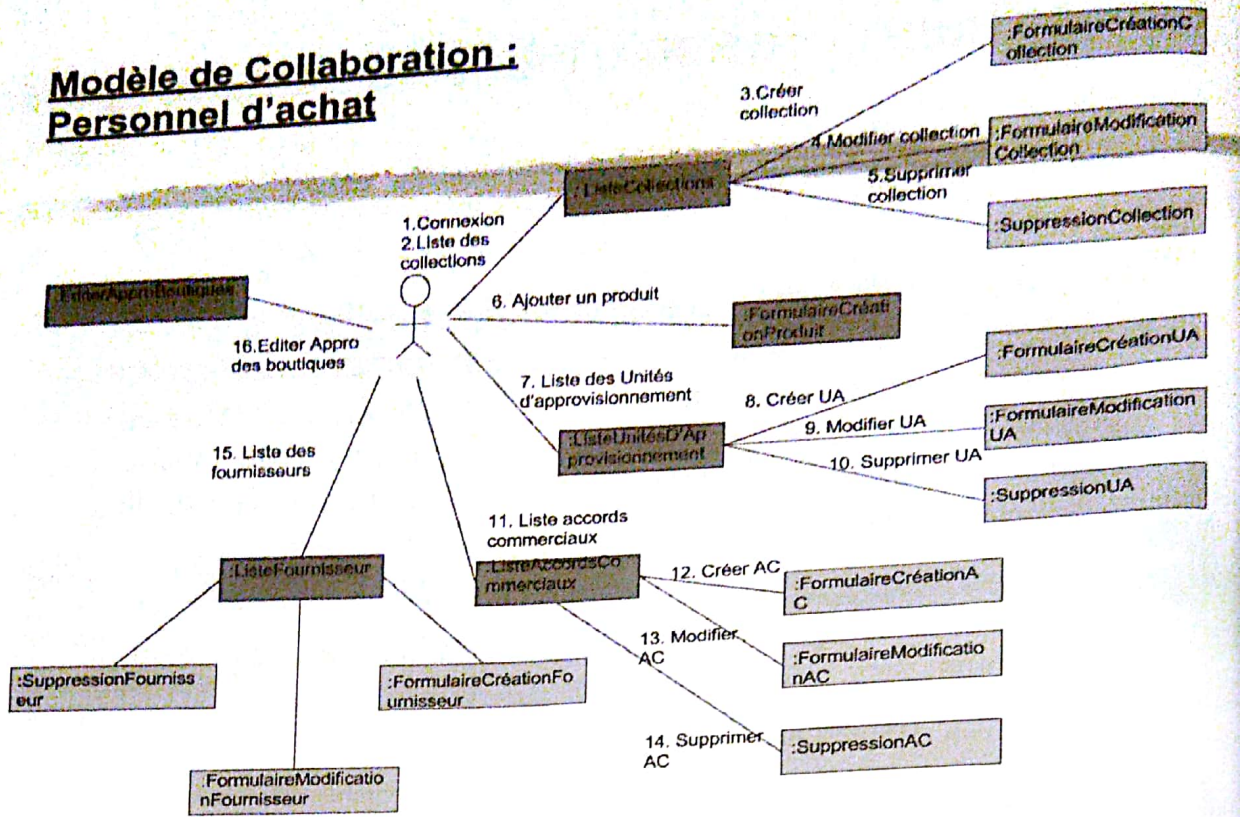
---

- ♣ LE DIAGRAMME DE COLLABORATION
- ♣ LES INTERACTIONS
- ♣ LES OBJETS
- ♣ LES MESSAGES
- ♣ EXEMPLES: GESTION DE COLLECTIONS et SERVICES BANCAIRES PAR INTERNET



71

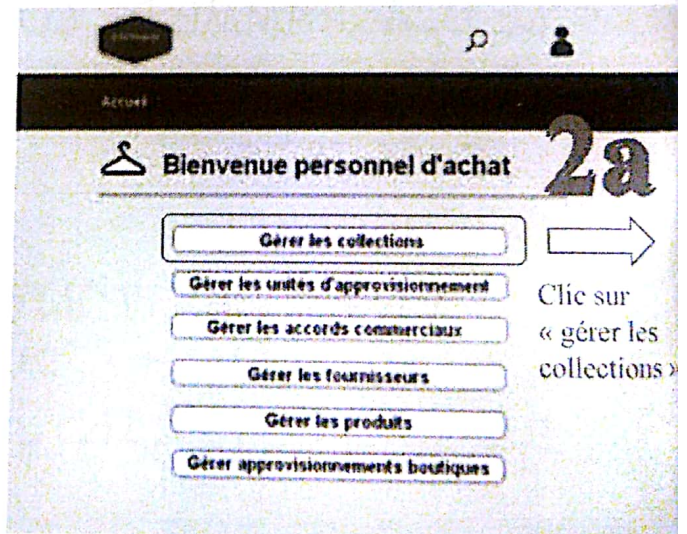
# Modèle de Collaboration : Personnel d'achat



© V. Henry

72

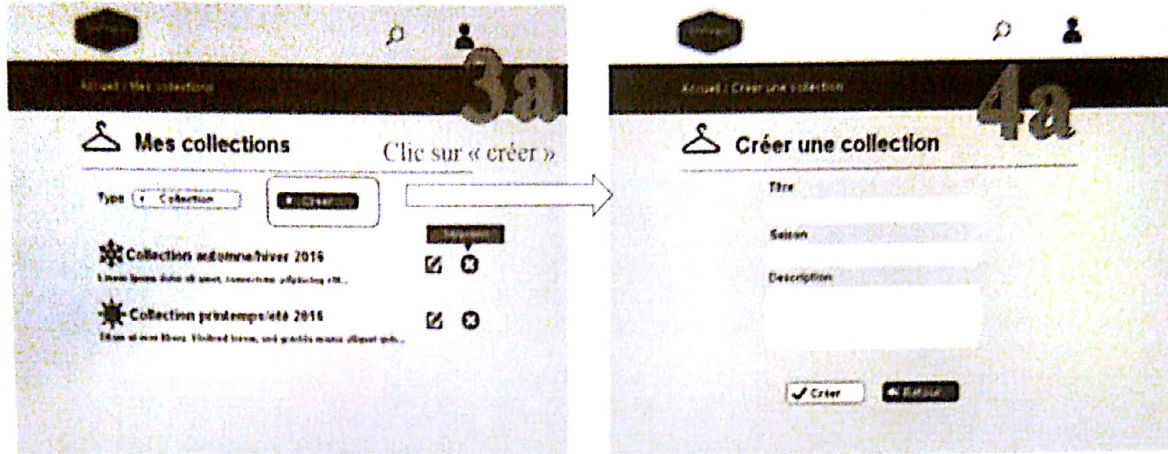
Exemple: formulaires de l'interface graphique



© V. Henry

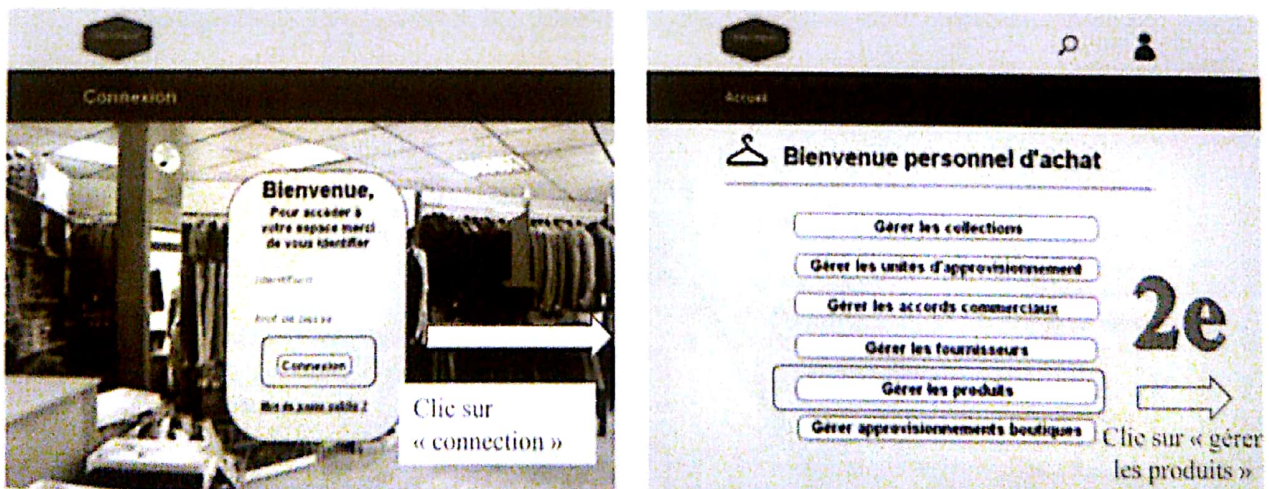
73

# Exemple: formulaires de l'interface graphique



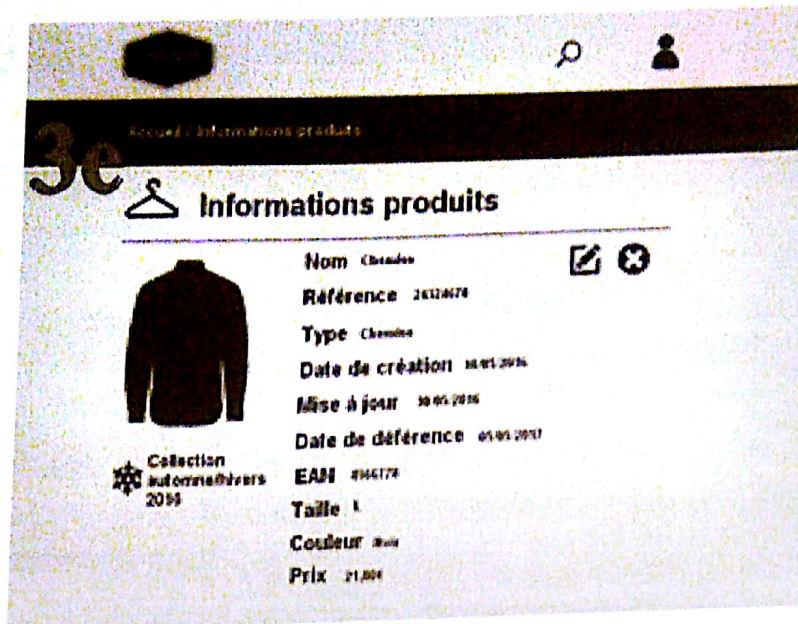
@V. Heiwy

# Exemple: formulaires de l'interface graphique



@V. Heiwy

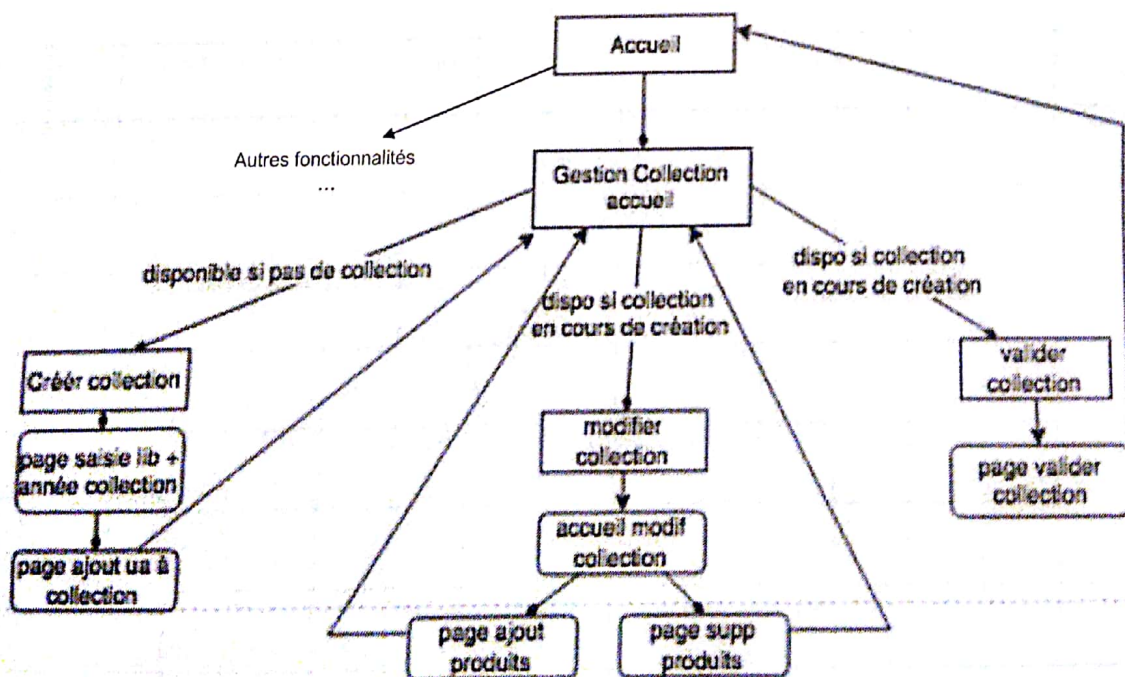
# Exemple: formulaires de l'interface graphique



@V. Heiwy

76

# Exemple :Echânement d'écrans



@V. Heiwy

77

## LE DIAGRAMME DE COLLABORATION (DCO)

- ❖ Diagramme d'interactions, variante du diagramme de séquences.
- ❖ Il met en évidence les interactions entre les objets du système étudié.
- ❖ Il précise le contexte d'évolution de chaque objet.
- ❖ Il montre les dépendances entre les objets impliqués dans l'exécution du processus ou cas d'utilisation.
- ❖ Il fait apparaître les **INTERACTIONS** entre **OBJETS** et les **MESSAGES** échangés.
- ❖ Il peut être utilisé pour représenter un **contexte**, une **organisation** ou des **enchaînements d'écrans**

78

## LE DIAGRAMME DE COMMUNICATION *Interaction et Objet*

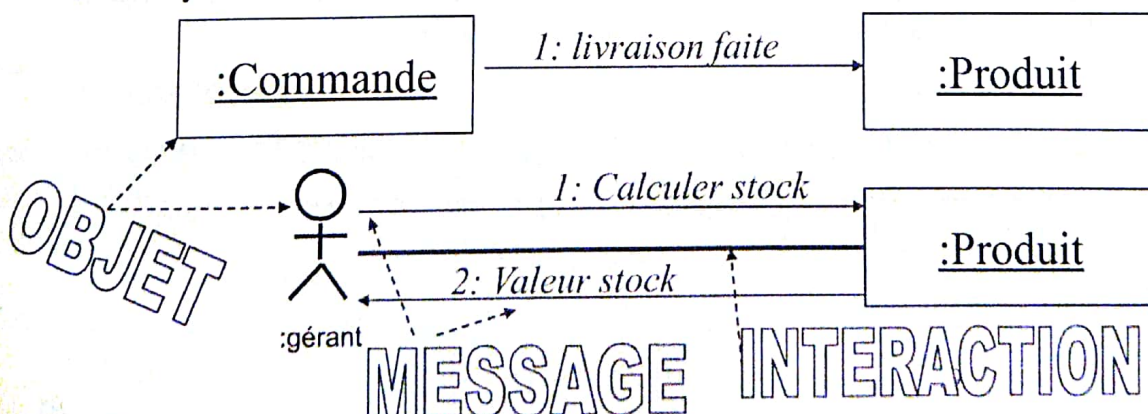
### Interaction

- ❖ L'**OBJET EMETTEUR** envoie un **MESSAGE** à un **OBJET RECEPTEUR**, suite à un **EVENEMENT**.

### Objet

- ❖ L'**OBJET** est soit une **instance de classe**, soit un **acteur**.

### Exemples:



79

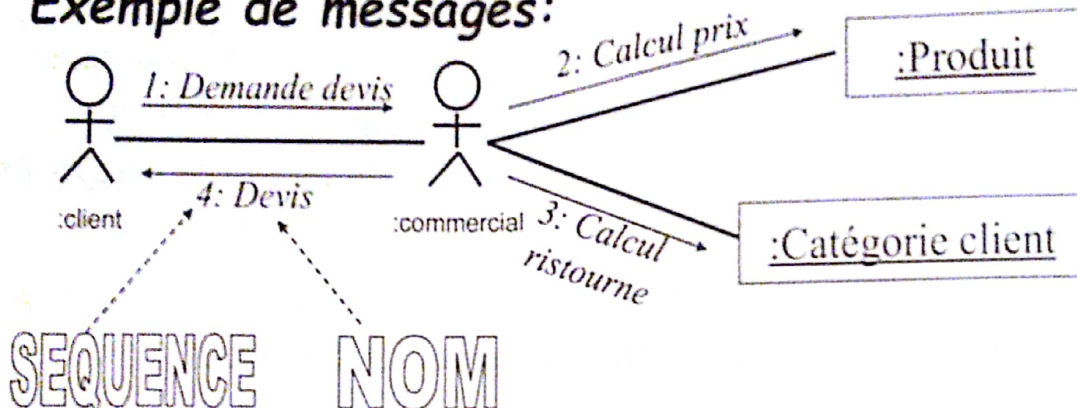
# LE DIAGRAMME DE COMMUNICATION

## Message (1)

### Messages

- ✦ Moyen de communication entre objets.
- ✦ Il peut être décrit par un nom, une séquence, des arguments, un résultat attendu, une synchronisation, une condition d'émission.

### Exemple de messages:



80

# LE DIAGRAMME DE COMMUNICATION

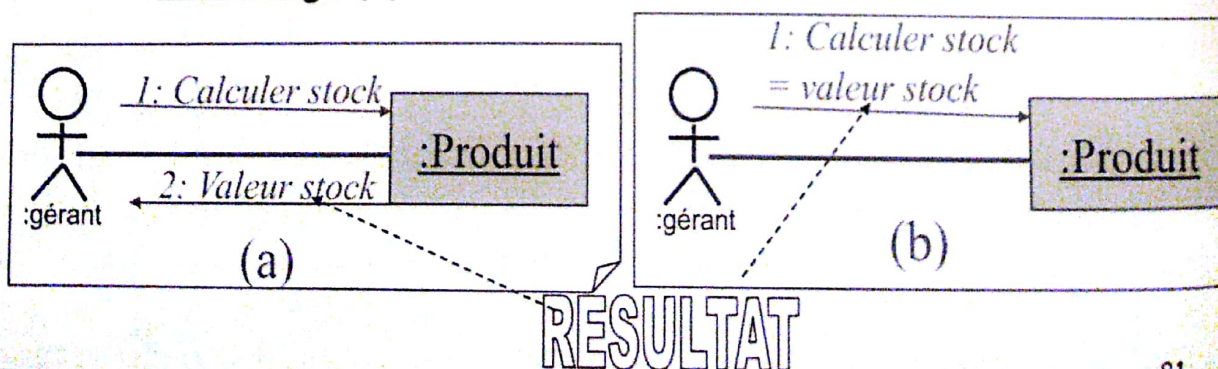
## Message (2)

### Exemple d'Argument

- La catégorie client
- L'intitulé du produit souhaité

### Exemple de résultat

En général il est représenté en utilisant un deuxième message (a). Pour simplifier, on le représente souvent dans le message (b).



81

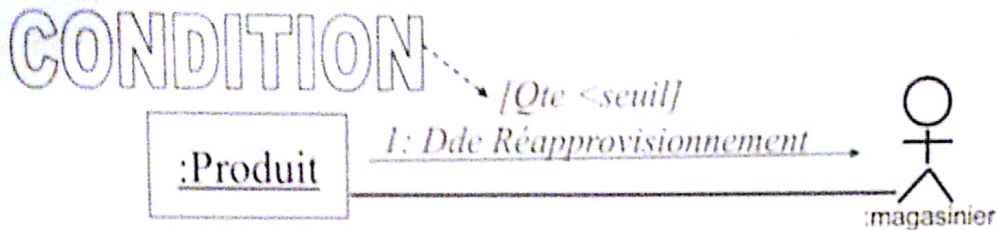
# LE DIAGRAMME DE COMMUNICATION

## Message (3)

### Exemple de Synchronisation

Associer l'ensemble des messages dans un message global unique.

### ✦ Exemple de Condition



- ✦ La demande de réapprovisionnement n'a lieu que si la quantité en stock est inférieure au seuil de réapprovisionnement.

82

82

# LE DIAGRAMME DE COMMUNICATION

## Message (4)

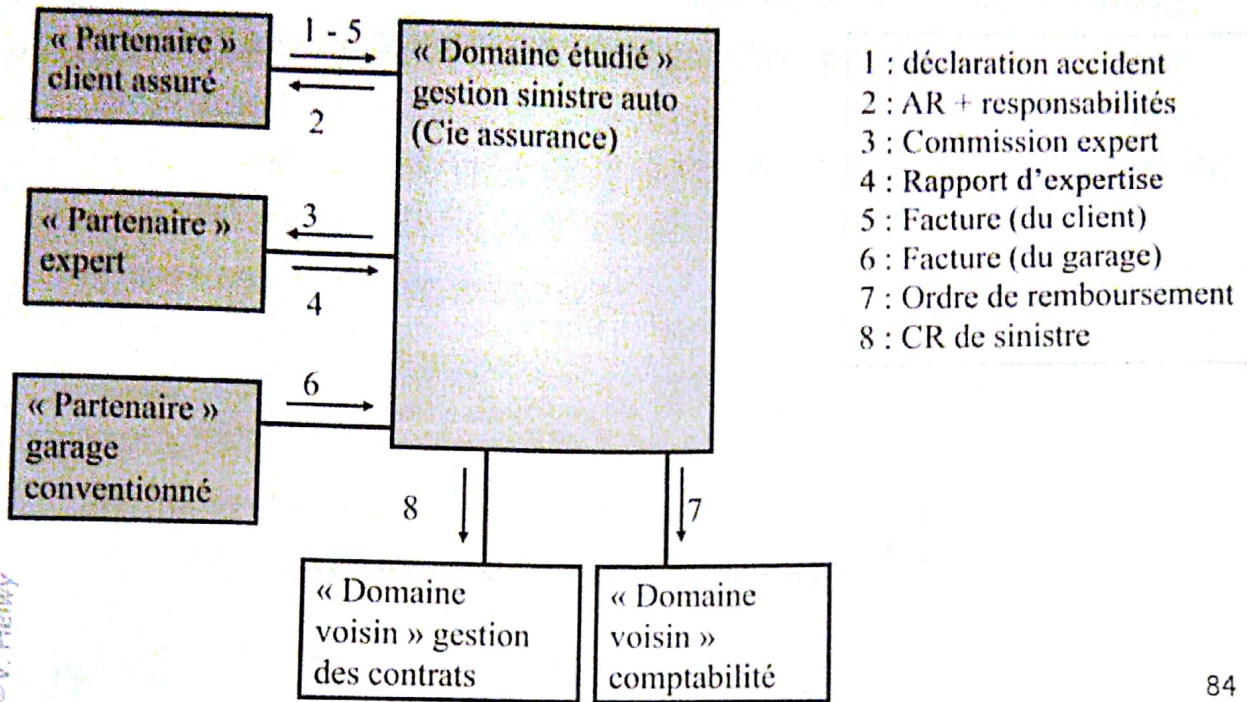
- ✦ **RESPONSABILITE** de l'acteur: prise en compte des différents messages.
- ✦ L'ensemble de ses responsabilités est défini par son **INTERFACE**.
- ✦ Un message peut être **REFLEXIF**. Dans ce cas l'objet émetteur est également l'objet récepteur.
- ✦ Exemple:



83

# Diagramme de communication

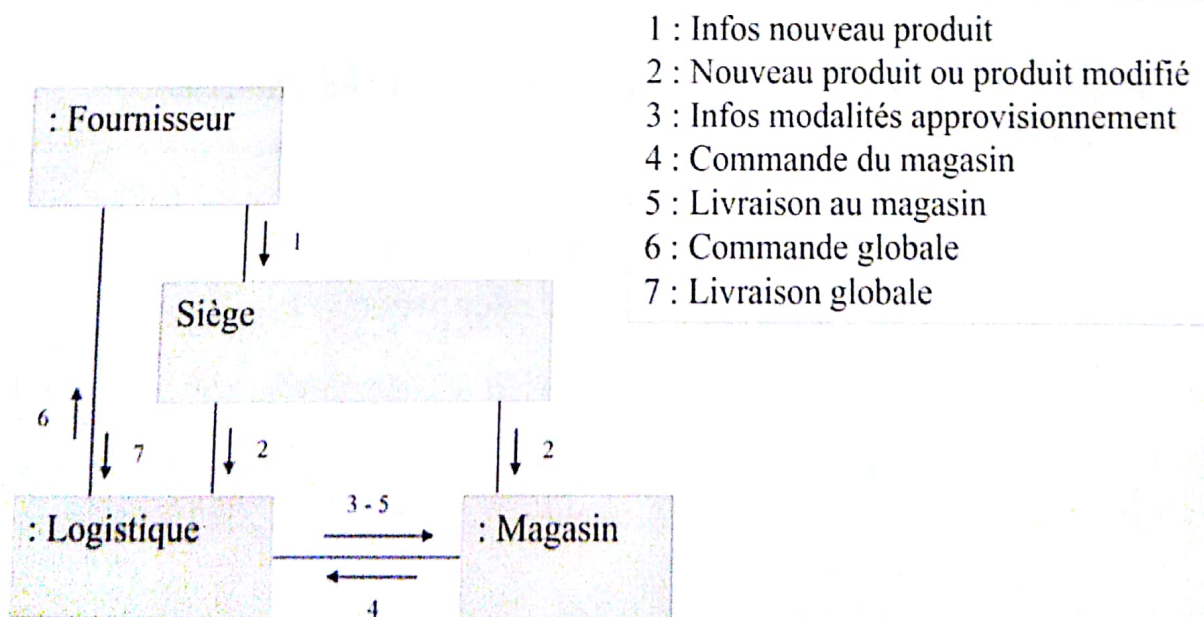
## Exemple : description d'un contexte



# Diagramme de communication

## Exemple : description d'une organisation

### ♣ Gestion des produits dans la grande distribution

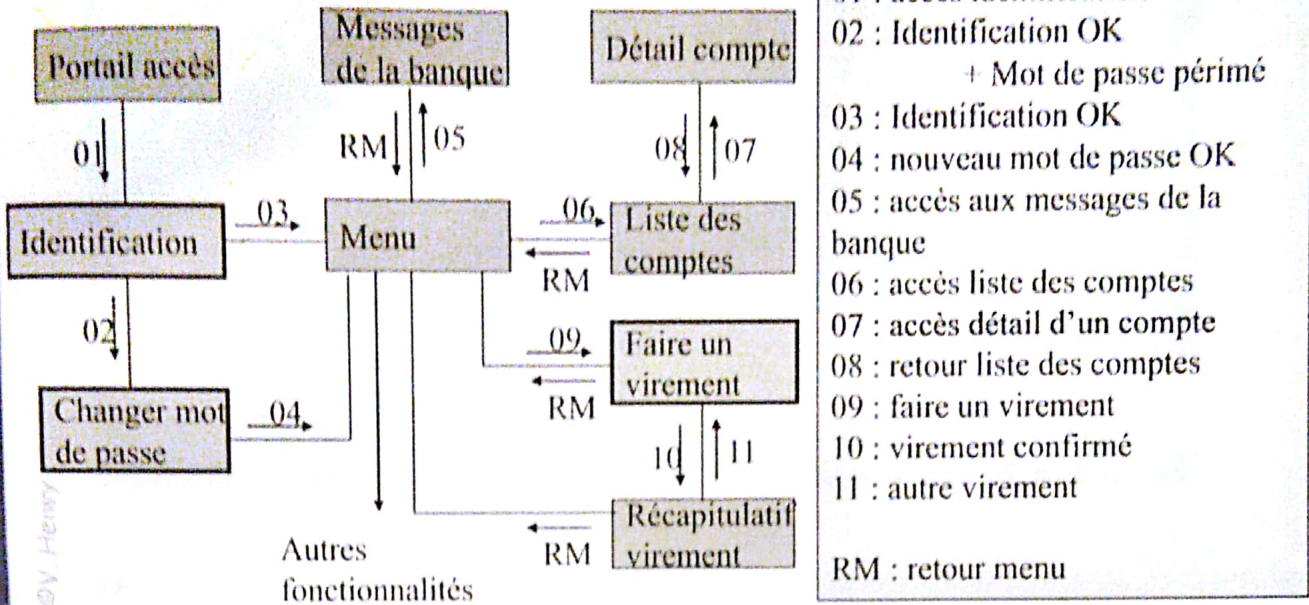


# Diagramme de communication

## Exemple : interactions dans une interface graphique

- Un diagramme de communication est utile pour représenter des enchaînements d'écrans ou fenêtres (logiques) :

ex : service bancaire par Internet

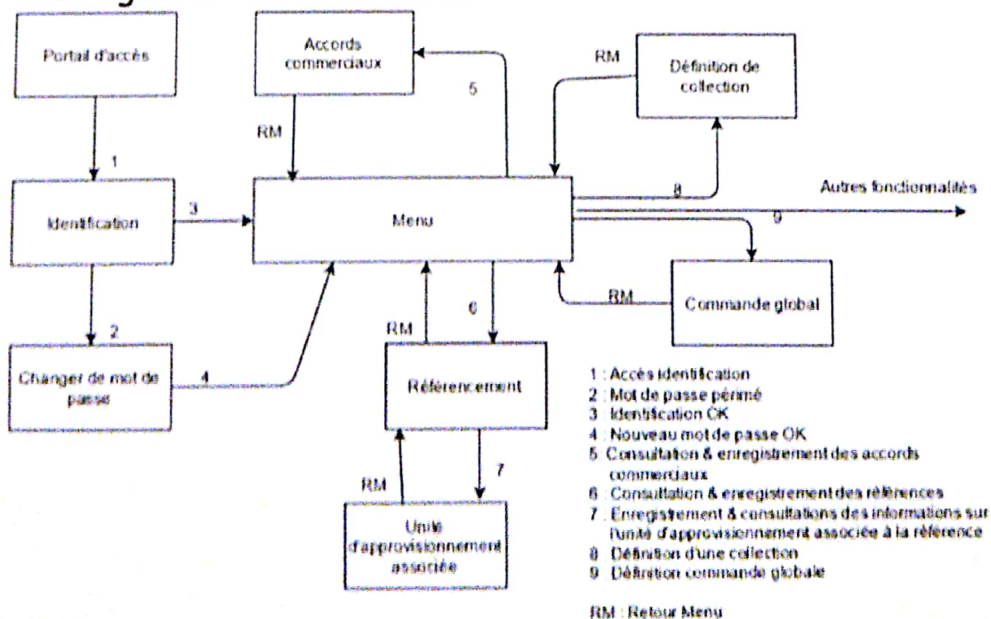


# Diagramme de communication

## Exemple : interactions dans une interface graphique

- Un diagramme de communication est utile pour représenter des enchaînements d'écrans ou fenêtres (logiques) :

ex : gestion des collections



Exemple maquette de l'interface associée au diagramme de communication

Exemple : gestion des collections

