

UE projet de développement

Licence 3 SPI, parcours Informatique

IHM - GUI

2022-2023

Intervenants : Paul-Antoine Bisgambiglia – Marie-Laure Nivet – Evelyne Vittori





Avant d'aller plus loin...

Un petit questionnaire : canal “Projet programmation IHM”

Teams L3SINF



IHM ? GUI ?

Objectifs généraux de l'UE

- Acquérir la compréhension et la maîtrise des concepts propres à la réalisation d'IHM
 - Séparation logique applicative/interface
 - Compréhension et pratique de la programmation évènementielle
- Développer en trinômes une application possédant une IHM
- Travailler l'expression écrite et orale
 - Présentation des rendus intermédiaires
 - Reviews de code
 - Présentation/démonstration finales



Présentation des principes du projet de développement

Ce qu'on attend de vous !

Principes/Impératifs du projet

- Travail en trinômes
- Application Desktop/Web possédant une interface graphique
- Conception itérative de la réalisation (UML, MCD, Maquettes, etc.)
- Pratique de la programmation événementielle
- Utilisation **collaborative** d'un dépôt Git partagé
- Choix libre du langage

Déroulé

- Points cours
- Réunions/points hebdomadaires
- Soutenances intermédiaires
- Soutenance finale en amont du départ en stage

Attendus - Evaluations

- Note Suivi Projet, travail au long court (30%)
 - Sérieux et implication dans le projet pendant tout son déroulé
 - Regularité du travail / Nombreuses itérations de Conceptions/Modélisations
 - Ponctualité des rendus
 - Qualité et soin des présentations et soutenances intermédiaires
 - Vie du dépôt Git
 - Qualité – vie du travail collaboratif
 - Prise en compte des retours
- Oral et démo de présentation finales, rapport de présentation finale : 20%
- Qualité – structuration du code – lisibilité du code : 20%
- Qualité – IHM finale : 30%

Objectifs – déroulé de cette séance

- Rapide histoire des IHM
- Les différents modes d'interaction avec la machine
- Présentation des principes de la programmation événementielle

A black and white photograph showing a man in a light-colored shirt working on a large, complex computer mainframe. He is looking down at the equipment. In the background, other people are visible, also working on similar machinery. The setting appears to be a technical or industrial environment from the mid-20th century.

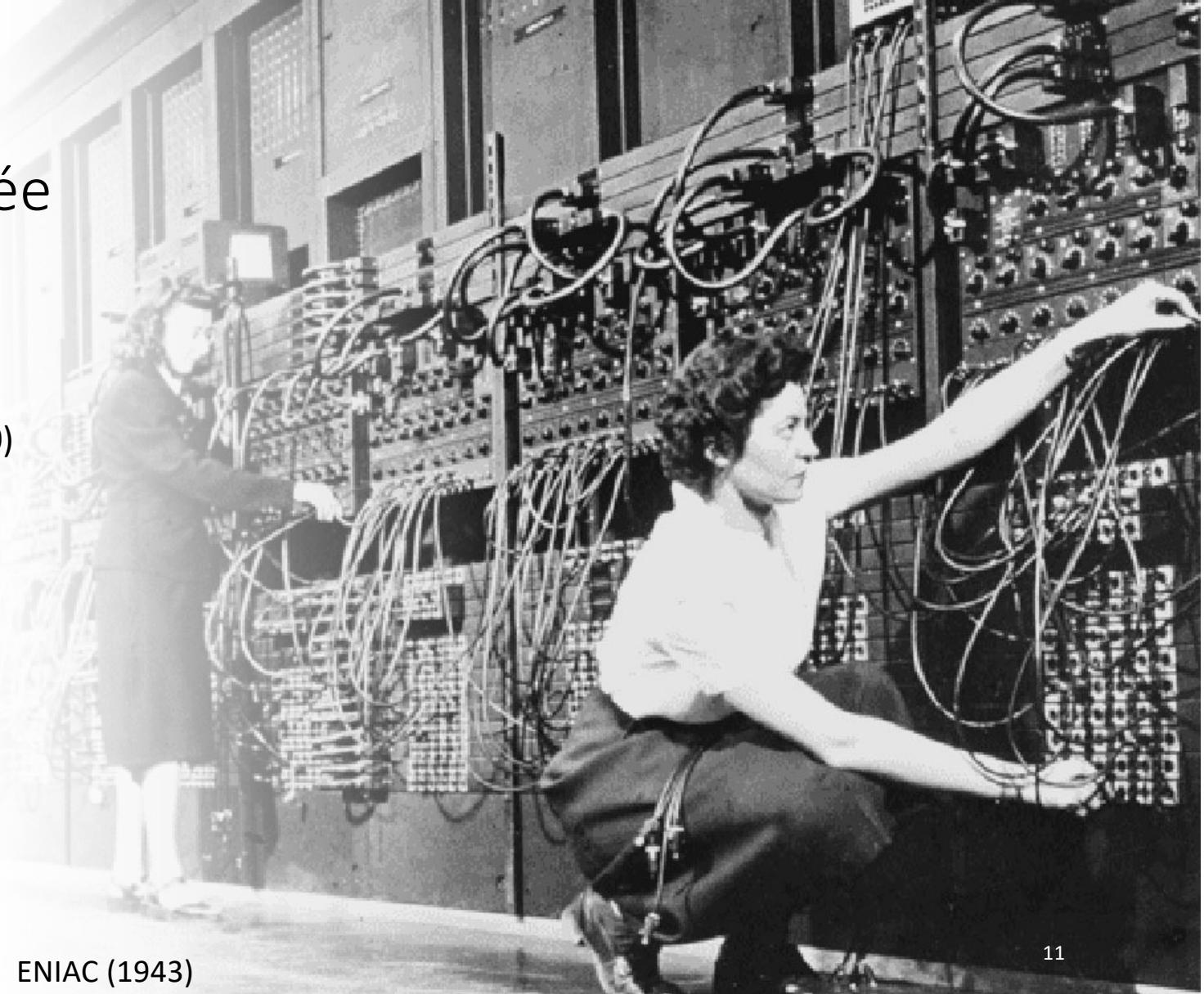
Historique des Interfaces Homme Machine

... Plus un petit regard vers le futur

Pour le passé : Computer History Archives Project (CHAP)
<https://www.youtube.com/@ComputerHistoryArchivesProject/videos>

Les débuts, une interaction limitée 1945 - 1965

- Fils/Jacks/interrupteurs
- Cartes perforées (INA 1960)
- Voyants, cadrant
- Imprimantes



L'arrivée des écrans/ claviers et les expériences de dispositifs de pointage



Sketchpad Demo 1963

https://www.youtube.com/watch?v=6orsmFndx_o

Ivan Sutherland's (1938 -) and Tim Johnson's (à l'image) Sketchpad (1963) 2 of 3 - 2D Graphics (9m56)

<https://www.youtube.com/watch?v=hB3jQKGrJo0>

Presentation du Sketchpad dans le cadre de la thèse de Sutherland MIT 1963 (16mn41)

<https://www.youtube.com/watch?v=57wj8diYpgY>

Les prémisses : Douglas Engelbart 1968

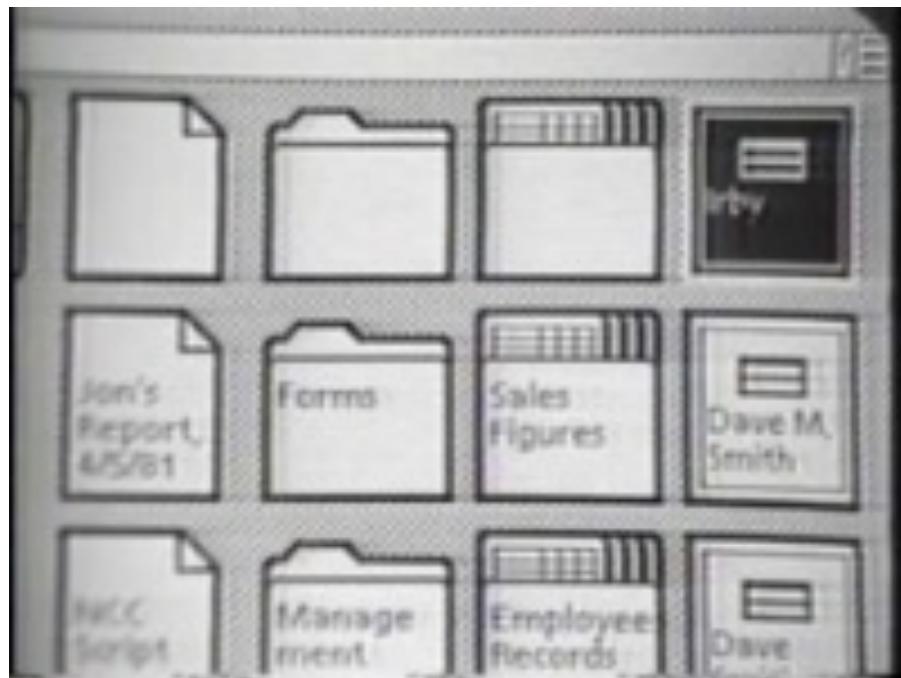
La première souris



The Mother of All Demos, presented by Douglas Engelbart (1968)

<https://www.youtube.com/watch?v=yJDv-zdhzMY>

Xerox et la première GUI (années 1980)



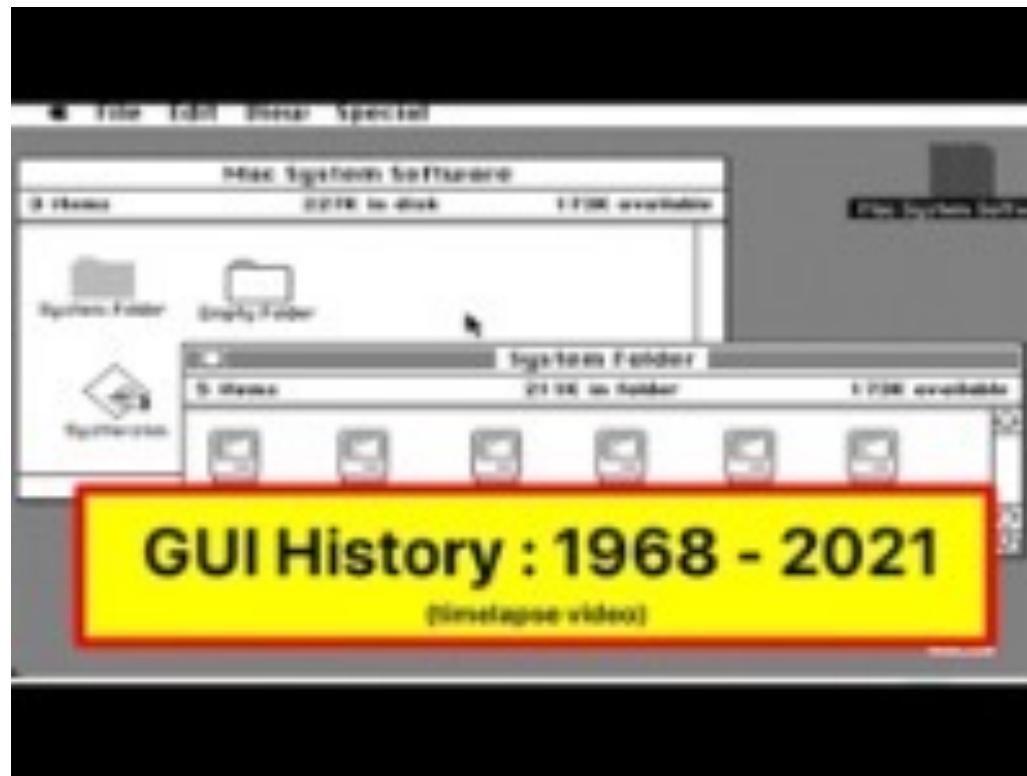
Xerox Star User Interface (1982) 1 of 2

<https://www.youtube.com/watch?v=Cn4vC80Pv6Q>

How Steve Jobs got the ideas of GUI from XEROX

<https://www.youtube.com/watch?v=J33pVRdxWbw> (7mn00)

History of GUI (1968-2020) | Apple | Microsoft | Xerox & More



Les différents types d'interfaces hommes machines

De l'interface textuelle, aux interfaces cerveau ordinateur...

Interfaces textuelles, *Command Line-oriented Interfaces (CLI)*

- Interfaces à ligne de commandes, *Command Line Interface*
 - Dialogue questions/réponses fortement structuré
 - Construit sur des langages de commandes
 - Effort de mémorisation important
 - Saisie par le clavier (*character-oriented interface*)

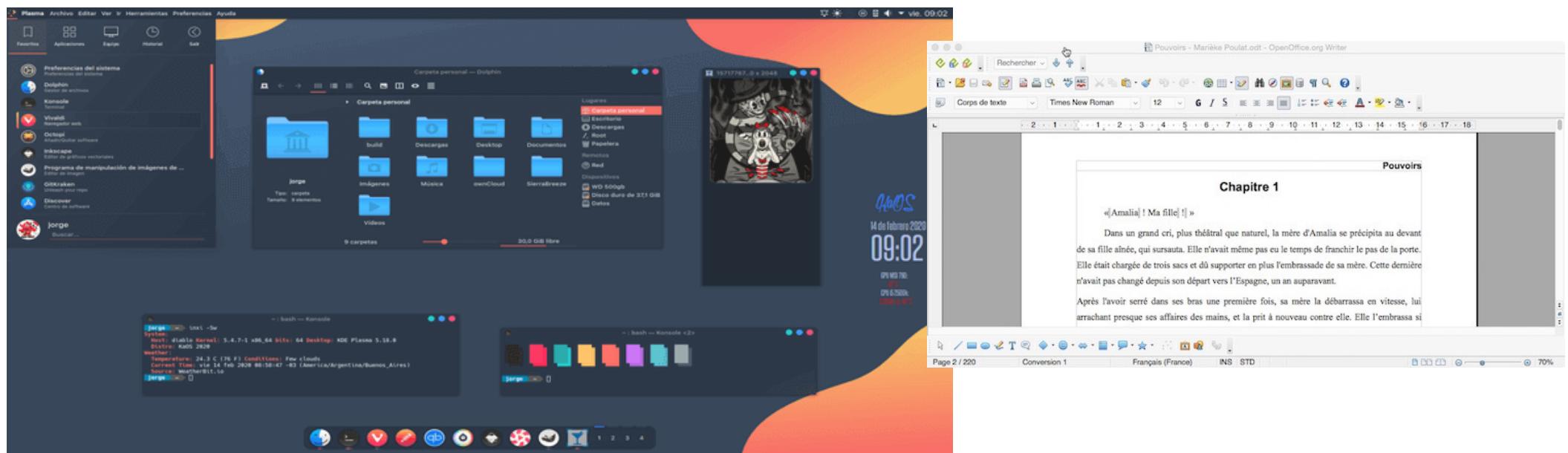
```
[root@localhost ~]# ping -q fa.wikipedia.org
PING text.mptp.wikimedia.org (208.80.152.2) 56(84) bytes of data.
PC
--- text.mptp.wikimedia.org ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 540.528/540.528/540.528/0.000 ms
[root@localhost ~]# pwd
/root
[root@localhost ~]# cd /var
[root@localhost var]# ls -la
total 72
drwxr-x--x 18 root root 4096 Jul 30 22:42 .
drwxr-x--x 23 root root 4096 Sep 14 20:42 ..
drwxr-x--x 2 root root 4096 May 14 00:15 account
drwxr-x--x 11 root root 4096 Jul 31 22:26 cache
drwxr-x--x 3 root root 4096 May 18 16:03 db
drwxr-x--x 3 root root 4096 May 18 16:03 empty
drwxr-x--x 2 root root 4096 May 18 16:03 games
drwxrwx-T 2 root gdm 4096 Jun 2 18:39 gdm
drwxr-x--x 38 root root 4096 May 18 16:03 lib
drwxr-x--x 2 root root 4096 May 18 16:03 local
lnwxrwxnx. 1 root root 11 May 14 08:12 lock -> ../run/lock
drwxr-x--x 14 root root 4096 Sep 14 20:42 log
lnwxrwxnx. 1 root root 10 Jul 30 22:43 mail -> spool/mail
drwxr-x--x 2 root root 4096 May 18 16:03 nis
drwxr-x--x 2 root root 4096 May 18 16:03 opt
drwxr-x--x 2 root root 4096 May 18 16:03 preserve
drwxr-x--x 2 root root 4096 Jul 1 22:11 report
lnwxrwxnx. 1 root root 6 May 14 00:12 run -> ../run
drwxr-x--x 14 root root 4096 May 18 16:03 spool
drwxrwxwt 4 root root 4096 Sep 12 23:56 tmp
drwxr-x--x 2 root root 4096 May 18 16:03 yp
[root@localhost var]# yum search wiki
Loaded plugins: langpacks, presto, refresh-packagekit, remove-with-leaves
rpmlfusion-free-updates | 2.7 kB 00:00
rpmlfusion-free-updates/primary_db | 206 kB 00:04
rpmlfusion-nonfree-updates | 2.7 kB 00:00
updates-metainfo | 5.9 kB 00:00
updates | 4.7 kB 00:00
updates/primary_db | 73% [=====] 62 kB/s 2.6 MB 00:15 ETA
```

Toujours très efficaces dans le cadre d'un usage intensif (rapidité)

- Autocad Raccourcis/langage de commande

Les interfaces graphiques, Graphical User Interfaces (GUI)

- Actions directes sur les objets apparaissant à l'écran
- Concept « What You See Is What You Get » (WYSIWYG)



Les interfaces intuitives, Natural User Interfaces (NUI)

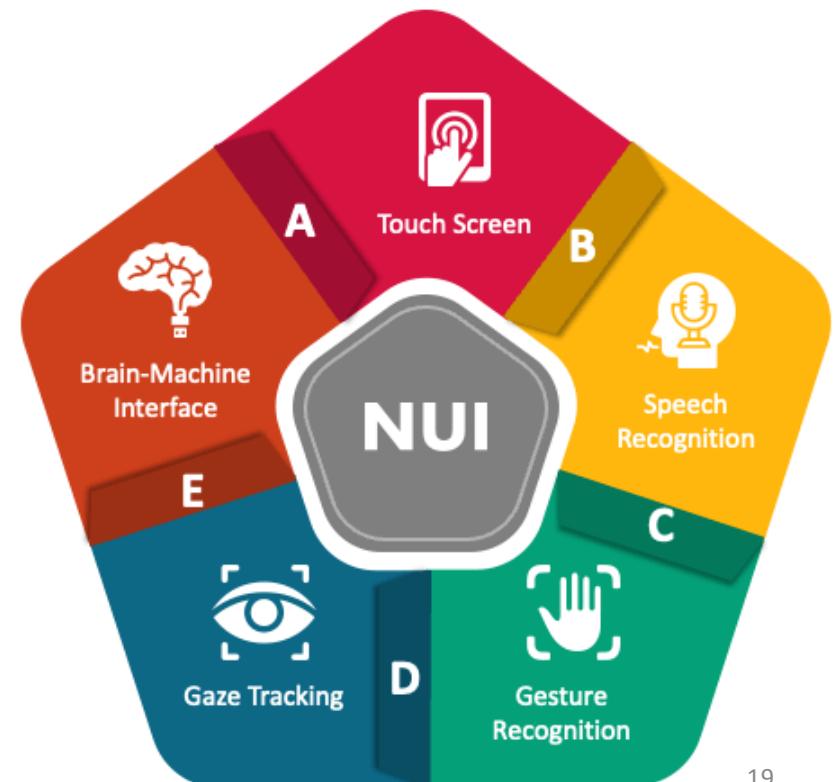
- Interaction naturelle, prise en mains directe.
- Commandes vocales, gestuelles, multimodales, pensée... !.



Voice User Interfaces

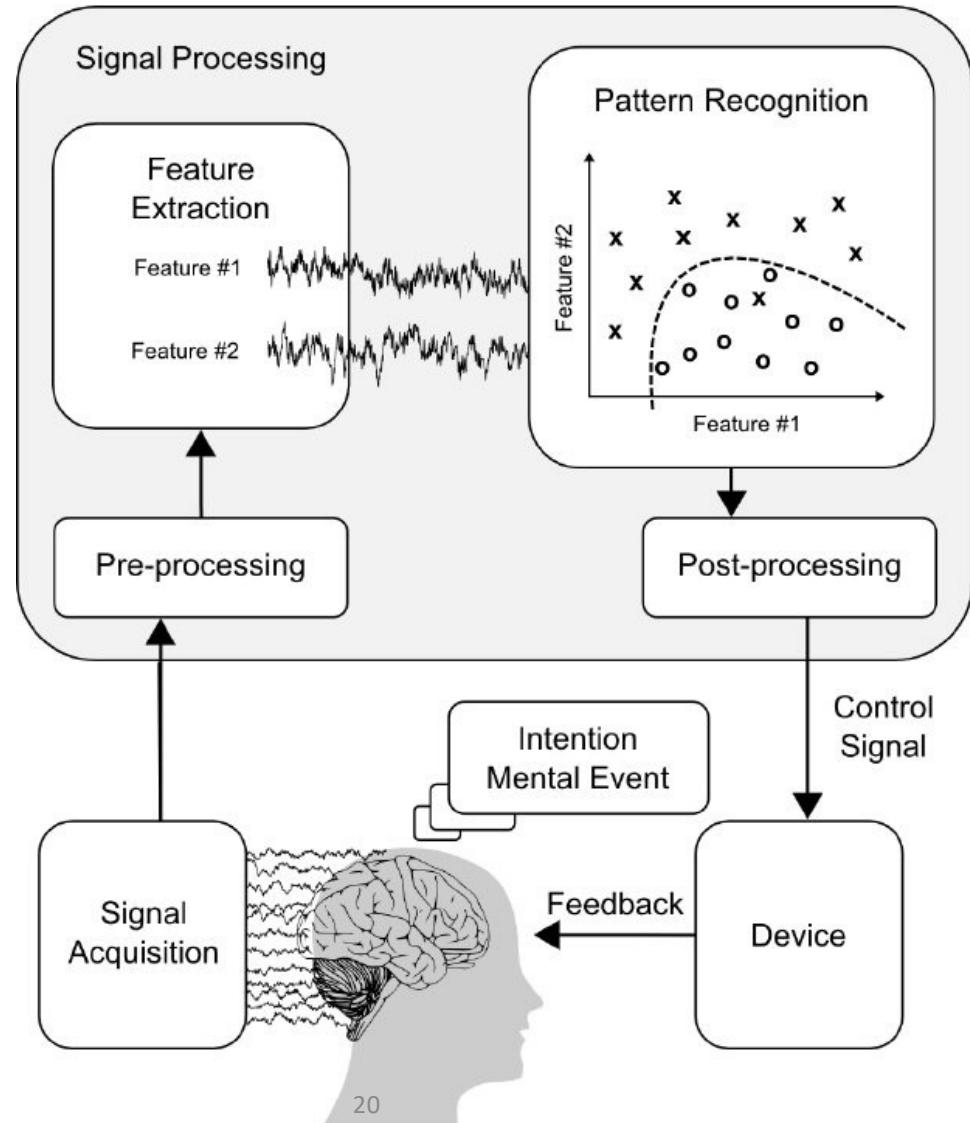
NATURAL USER INTERFACE (NUI)

Elements of Natural User Interface



Interfaces cerveau ordinateurs, *Brain Computer Interfaces* (BCI)

- Intérêt : grande vitesse de réaction,
peu de latence entre l'événement et
- ... A finir



BCI



Brain Computer Interface (VOSTF) INRIA, 2017



Immersion, XR, *Extended Reallity*

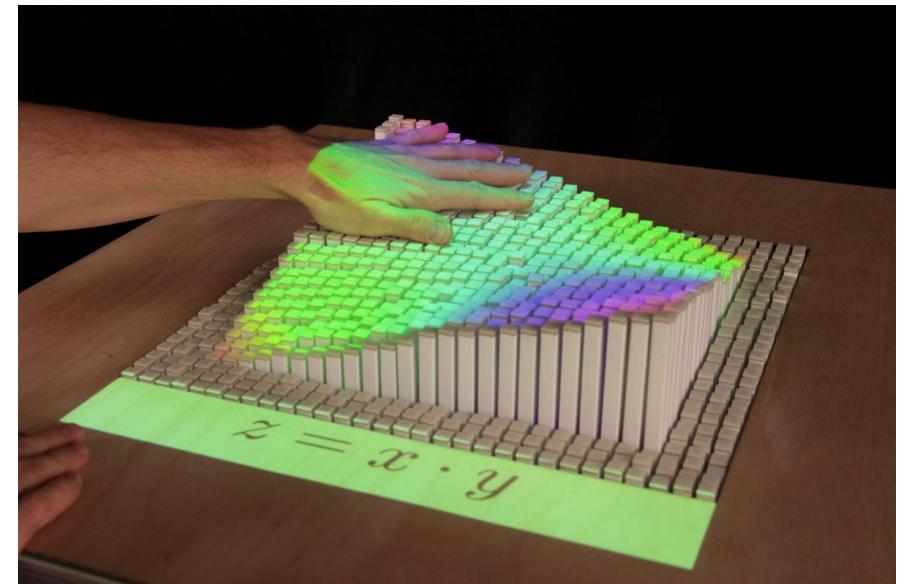
- Réalité virtuelle
- Réalité augmentée
- Réalité mixte

Interfaces tangibles, Tangible User Interfaces (TUI)

- Interaction avec un ou des objets tangibles, réels qui permettent d'agir sur des objets numériques



En entrée



En sortie

Principaux composants d'une interface graphique Application Desktop

- Un pointeur/curseur utilisé pour la sélection et la manipulation
 - Flèche en mode graphique,
 - Curseur clignotant en mode textuel
- Un dispositif de pointage
 - Track ball
 - Souris
- Des icônes
- Un bureau
- Des fenêtres
- Des menus

Fondamentaux des interfaces homme machine

Qu'est-ce qu'une interface homme machine ?

- Ensemble des dispositifs matériels et logiciels qui permettent à un utilisateur humain d'interagir avec un système numérique interactif



Généralités

Lorsque l'on conçoit une Interface Homme Machine (IHM) il ne faut pas oublier que le partenaire principal est l'homme et non la machine !

Les domaines entrant en ligne de compte

- Informatique : la machine, le langage, l'algorithme
- Sciences cognitives : prise en compte de l'être humain, de ses biais
- Ergonomie : l'interaction entre l'homme et la machine
- Sociologie, psychologie : les émotions participent de notre réceptivité.

Les participants à la réalisation d'une interface

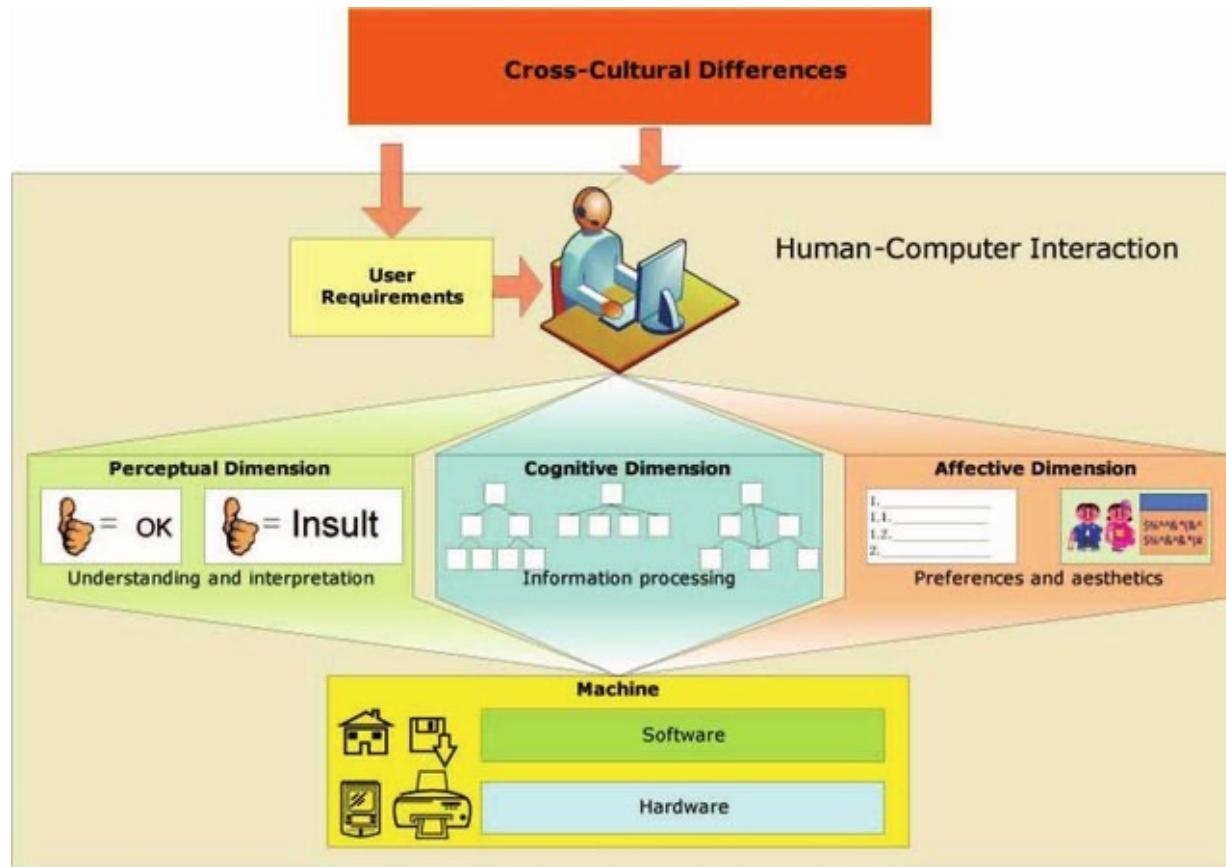
- Le futur utilisateur
 - Il expose ses besoins, ses choix
- La machine et plus particulièrement le programme à interfacer
- Le concepteur de l'interface
 - Censé coder – et donc connaître ! – les besoins et les choix de l'utilisateur

Adaptation de l'IHM

Prendre en compte :

- Les spécificités physiques
- L'expertise métier, l'expertise niveau manipulation
- Les potentielles spécificités culturelles
 - Formats de dates, de nombres
 - Signification des couleurs, de l'icônerie
 - Sens d'écriture
- Le contexte (grand public, industrie, loisir, système critique)
- Les contraintes techniques
 - Plateformes, Compatibilité Legacy, bande passante, mémoire, écran, etc.

Les principes généraux de l'interaction homme machine



B.A. B.A.

- Ne pas chercher à être avant tout innovant, nouveau, joli, cool...
- Faire simple mais clair !
- Utiliser les conventions propres au système auquel est destiné l'interface
- Avant d'imposer votre interface aux autres, regarder la bien et tester la vraiment...

Quelques conseils...

Des règles à respecter

Restez cohérents

- Cohérence externe

- dans le choix des symboles pour représenter les choses. Ex:
La poubelle pour jeter les document que l'on ne veut plus,
rouge pour attention, vert pour allez-y.
- choisir des termes, des images qui désignent bien ce qu'on
veut dire en évitant les termes généraux ou ambigus que
l'usager prendra du temps à interpréter dans le contexte.

- Cohérence interne

- les symboles , les dispositions doivent être constants.
- réutiliser le même mot pour désigner la même chose.

Adaptabilité à l'utilisateur

- Prévoir plusieurs niveaux d'utilisation
 - Débutants (Menus, barres d'outils)
 - Confirmés (raccourcis, personnalisation)
- Prévoir plusieurs niveaux de transparence
 - Cacher les mécanismes internes de l'appli
 - Permettre l'administration et la mise au point

Rétro-action (feedback)

- Visualisation de l'effet des actions
 - Le logiciel doit réagir pour montrer l'effet d'une action.
- Dans l'absence de rétroaction, l'usager ne sait pas s'il a bien choisi, ce qu'il a choisi, si la commande qu'il utilise est possible...
- Le système doit réagir de façon continue à ce que fait l'usager en l'informant des choix faits, des processus en cours.
 - Rapide et non ambiguë
 - Modification du curseur de la souris
 - Sélection en vidéo inverse
 - Boutons enfoncés

Avertissement de fin de tâche

- Fin de tâche de fond
 - Envoi de mail, de fax
 - Fin d'impression
- Fin de succession de tâches atomiques

Gestion des erreurs

- Types d'erreurs
 - Apprentissage empirique (droit à l'essai)
 - Erreur physique
 - Erreur logique ou d'interprétation
- Prévention des erreurs
 - Adapter l'affichage au contexte
 - Vérifier qu'une entrée est valide
 - Proposer des corrections
 - Rendre inactifs les items de menus interdits
 - Design (Taille et choix des icônes)
 - Bulles d'aide
- Traitement des erreurs

Messages d'erreurs

- Spécifiques et précis
 - Éviter les « Syntax error N° »
 - Préférer « Parenthèse manquante ligne 10 »
- Constructifs, indiquer les corrections possibles
- Positifs
 - éviter la condamnation : ILLEGAL, INVALIDE

Réversibilité des actions

- Possibilité de "défaire" ce qui a été fait : *undo*
 - Boutons annuler dans les boîtes de dialogue
 - Fonction d'annulation des actions
- Importante pour que l'usager n'aie pas peur d'essayer les fonctionnalités.
- Favorise l'exploration, l'essai erreur et permet à l'usager d'acquérir de l'expertise.
- La principale limite à l'utilisation des systèmes vient souvent du fait que les usagers n'ont pas assez exploré un logiciel pour en connaître tout le potentiel.

Laisser la maîtrise à l'utilisateur

- L'utilisateur pilote l'application
- Limiter les situations de guidage total
- Préférer le droit à l'erreur
- Meilleure appropriation

Ne tomber pas dans la tentation des Dark Pattern

The Nerdwriter

- Travail à faire : recensement des Dark Pattern que vous rencontrez sur votre chemin d'utilisateur : Fichier partagé sur Teams

<https://www.novaway.fr/blog/ui-ux-design/dark-pattern-experience-utilisateur>

Des initiatives de la tech

- « THE CRITICAL ENGINEERING MANIFESTO »

<https://criticalengineering.org/fr>

al | ca | cn | de | dk | en | es | fi | fr | gl | gr | he | it | jp | nl | pt | ru | sl | sr



The Critical Engineering Working Group
Berlin, October 2011–2021

Julian Oliver
Gordan Savicic
Danja Vasiliev

THE CRITICAL ENGINEERING MANIFESTO

0. L'ingénieur critique considère l'ingénierie comme le langage le plus percutant de notre époque modélisant nos manières de nous déplacer, communiquer et penser. Le rôle de l'ingénieur critique est d'étudier et d'exploiter ce langage – de montrer son influence.

1. L'ingénieur critique considère notre dépendance à la technologie comme un défi autant qu'une menace. Plus la dépendance est grande plus il importe d'en analyser le fonctionnement interne indépendamment des droits de propriétés et dispositions légales.

2. L'ingénieur est conscient que chaque avancée technologique met à mal notre héritage techno-politique.

3. L'ingénieur critique déconstruit et incite à la méfiance face aux expériences utilisateurs luxuriantes.

4. L'ingénieur critique ne se laisse pas impressionner par les contraintes de l'implémentation. Il en explore les méthodes d'influence et leurs effets spécifiques.

5. L'ingénieur critique reconnaît que chaque travail d'ingénierie formate l'utilisateur proportionnellement à la dépendance de ce dernier au travail d'ingénierie.

6. En décrivant les interrelations des dispositifs, corps, agents et forces des réseaux, l'ingénieur critique élargit le concept de 'machine'.

7. L'ingénieur critique observe l'espace entre la production et la consommation des technologies. En agissant rapidement, l'ingénieur critique peut déstabiliser cet espace provoquant des moments de déséquilibre et de déception.

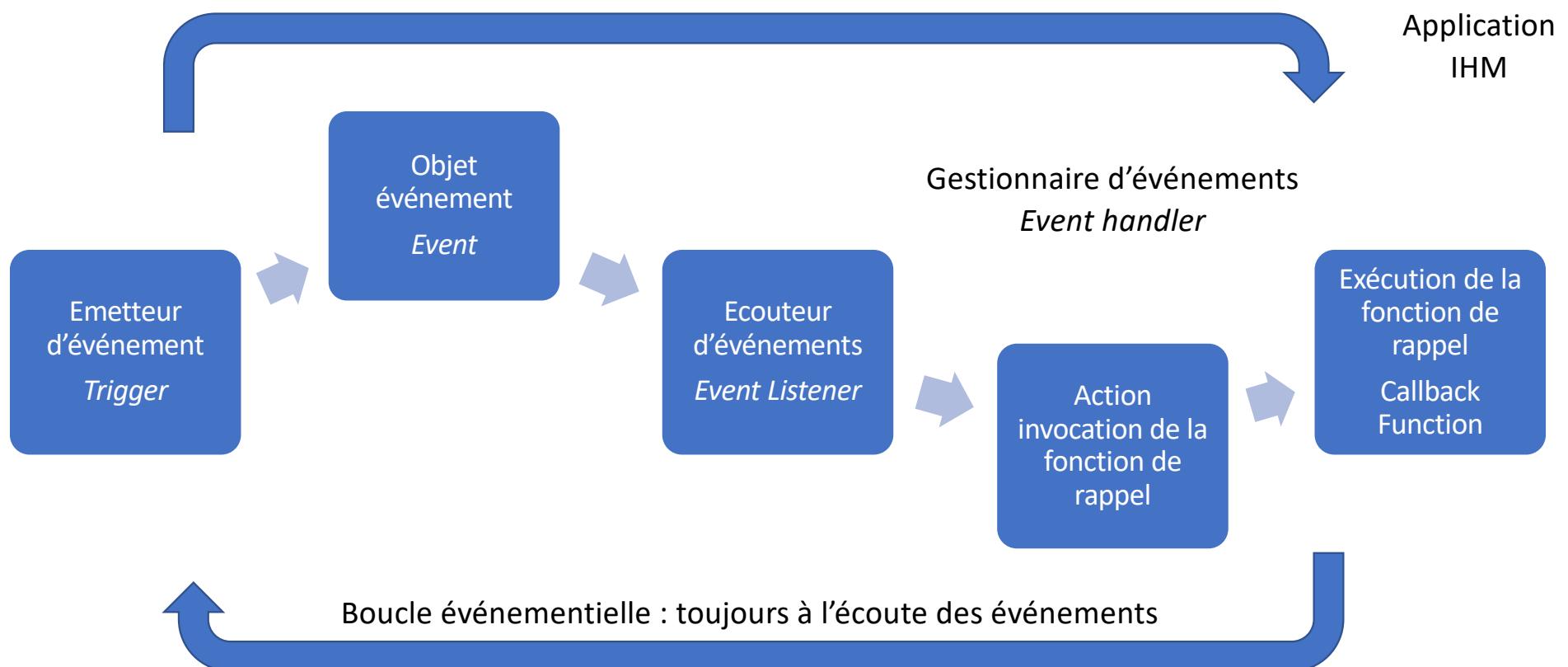
8. L'ingénieur critique se tourne vers l'histoire de l'art, l'architecture, l'activisme, la philosophie et l'innovation pour y trouver des travaux exemplaires d'ingénierie critique. Les stratégies, idées, agendas, de ces disciplines seront adoptées, re-agencées et développées.

9. L'ingénieur critique constate que le code écrit s'immisce dans les secteurs du social et du psychologique régulant les comportements entre les personnes et les machines avec lesquelles elles interagissent. Prenant conscience de ce fait, l'ingénieur critique cherche à reconstruire les contraintes et actions sociales des utilisateurs en procédant à des fouilles numériques.

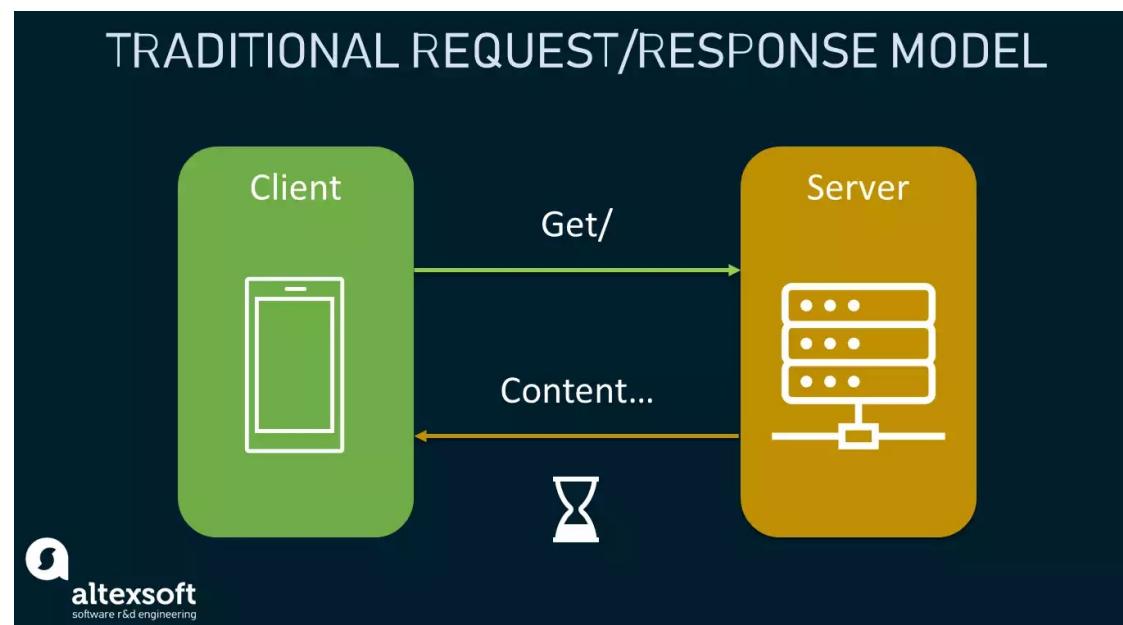
10. L'ingénieur critique considère l'utilisation de la vulnérabilité d'un système comme la forme la plus souhaitable de dénonciation.

Translation into French by [Benedicte Jacobs](#).

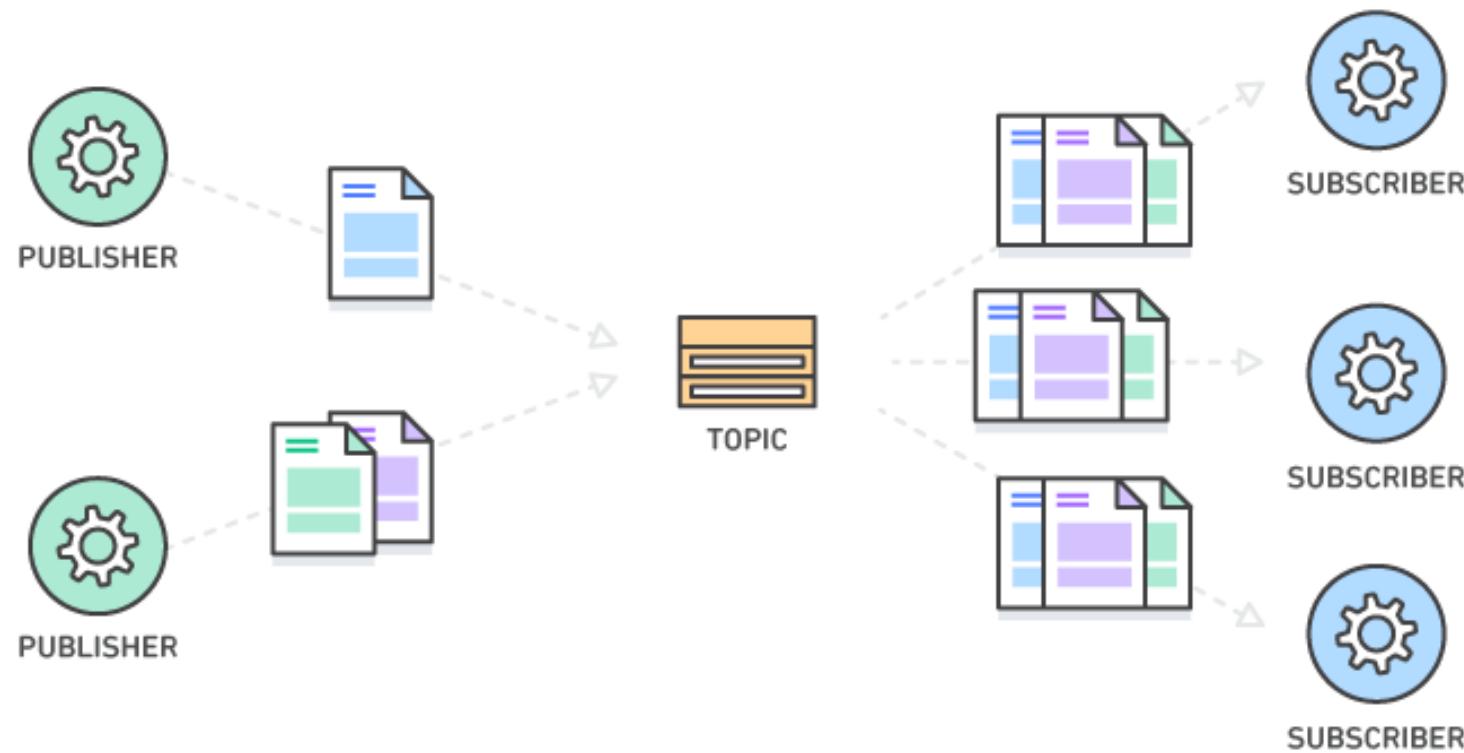
Modèle de la programmation événementielle

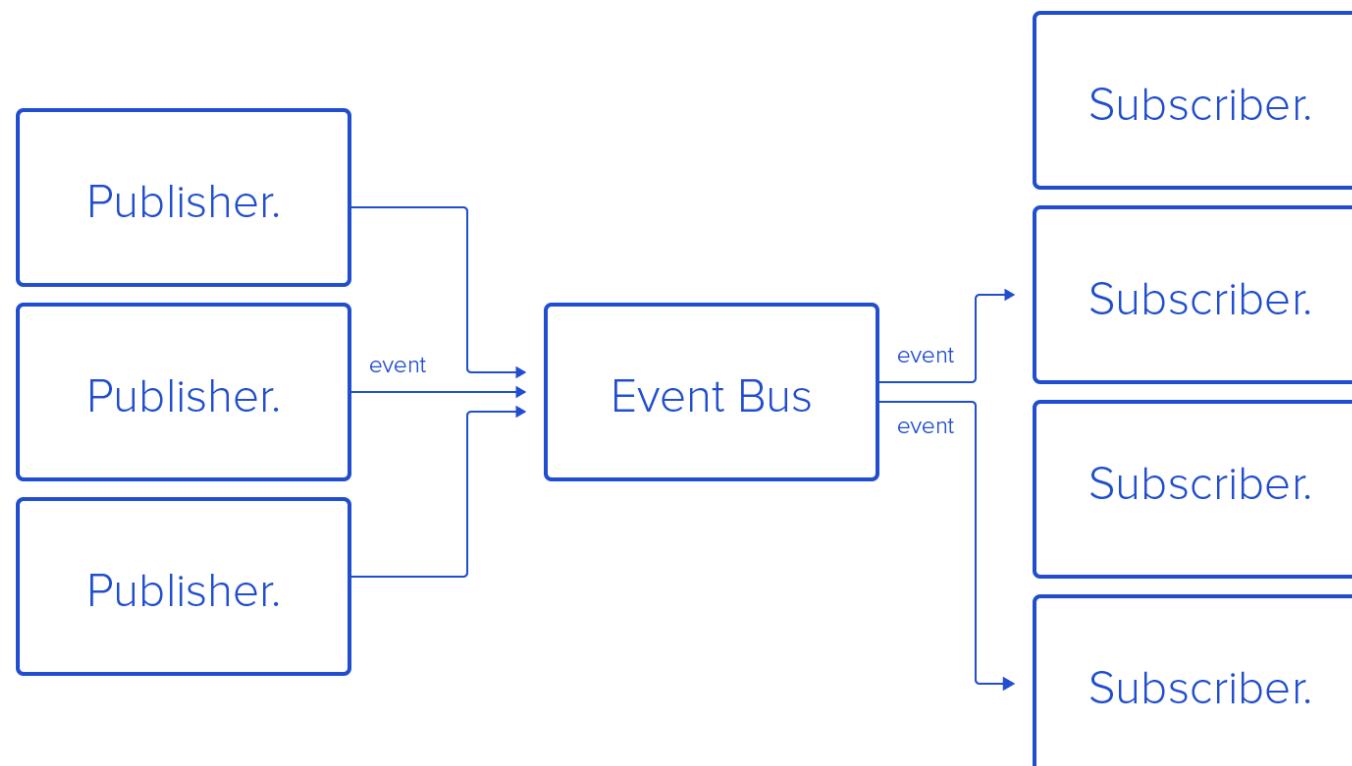


Modèle traditionnel Requête/réponse



Modèle Publish and Subscribe (Pubsub)





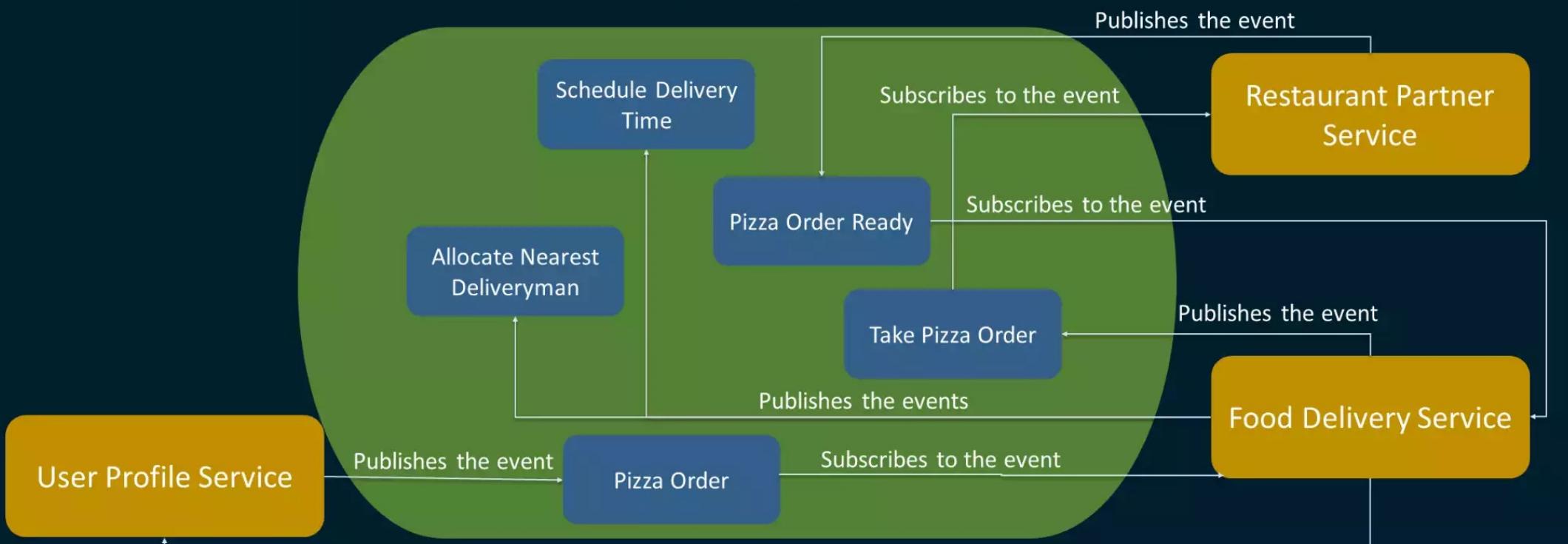
Modèle Publish and Subscribe (Pubsub)

EVENT-DRIVEN ARCHITECTURE COMPONENTS

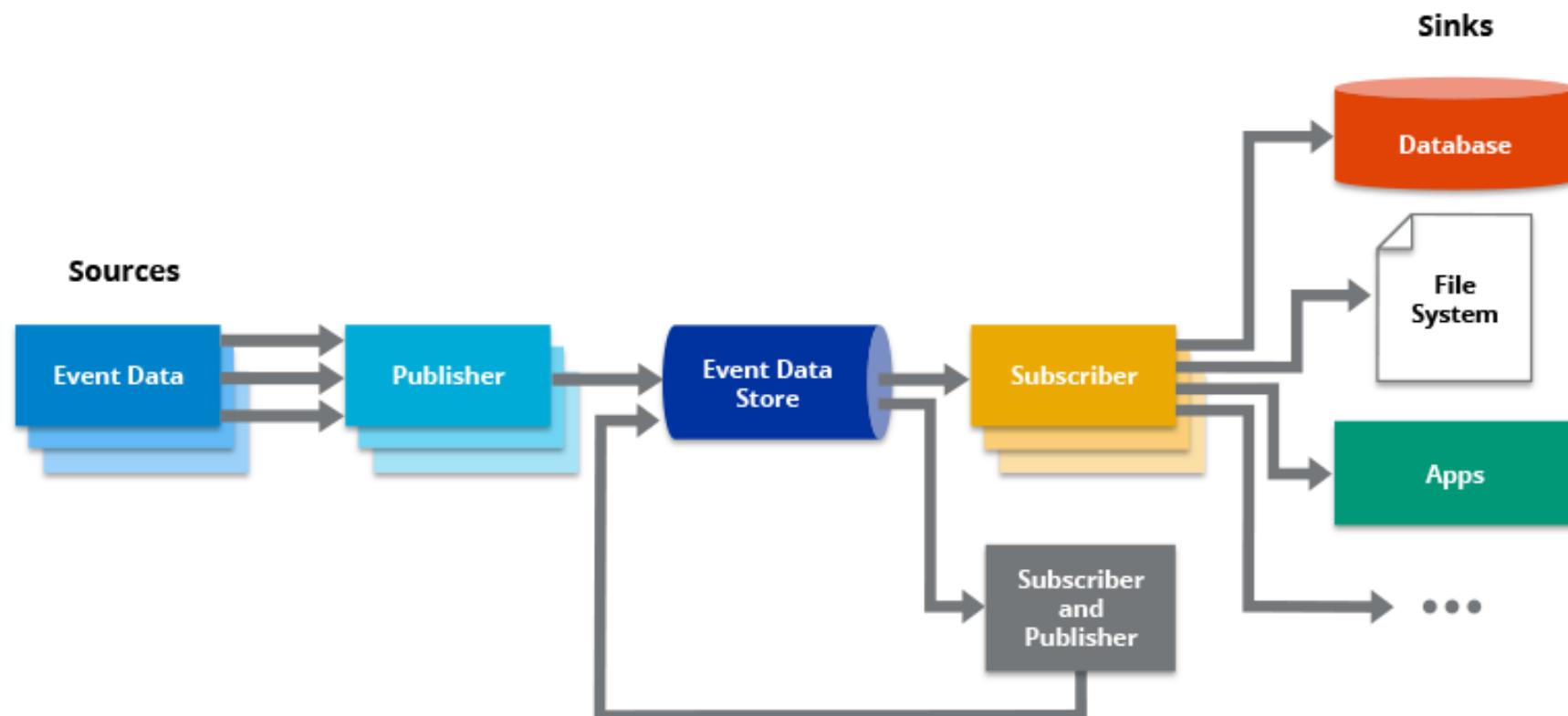


altexsoft
software r&d engineering

HOW THE PUB/SUB MESSAGING PATTERN WORKS (Food Ordering Scenario)



Modèle événementiel



Différents types d'événements

- Bloquants
- Non bloquants

- Etudes de cas : <https://design.cnil.fr/etudes-de-cas/fuzzy/>

Données & Design par LINC CNIL

CONCEPTS CLÉS ÉTUDES DE CAS DESIGN PATTERNS RESSOURCES ACTUALITÉS ENGLISH

Co-construire des parcours utilisateurs respectueux du RGPD et de la vie privée



Le design dans le RGPD

Approcher la réglementation par le design au-travers des concepts du RGPD activables par les designers.

[DÉCOUVRIR LES CONCEPTS CLÉS](#)



Des exemples concrets

Inspirez-vous des études de cas, co-construites avec la communauté, pour implémenter les concepts clés du RGPD dans vos services et produits numériques.

[VOIR LES ÉTUDES DE CAS](#)



Pour rester informé

Toutes les nouvelles sur les activités de Données&Design pour toujours être au courant des dernières publications et connaître nos prochains événements pour nous rencontrer.

[VOIR LES PROCHAINS ÉVÉNEMENTS](#)

Boucle événementielle

```
while True:
    # Get the next event from the operating system
    event = get_next_event()

    # Get the function that is assigned to handle this event
    a_function_to_handle_the_event = event-handlers[event]

    # If a function has been assigned to handle this event, call the function
    if a_function_to_handle_the_event:
        a_function_to_handle_the_event()  # Call the event-handler function

    # Stop processing events if the user gives a command to stop the application
    if window_needs_to_close:
        break  # out of the event-loop
```



Différents outils pour le design/prototypage d'interface homme machines

Figma, Balsamiq, Sketch, Canva, etc.

Tester une IHM

Les différentes méthodes de test, du manuel aux outils dédiés (Desktop ou Web)

Les différentes solutions de test

- Manuelle
- Outil d'automatisation : Enregistrer et rejouer
 - Unified Functional Testing (UFT, successeur de QTF)
 - Pour le Web (Selenium IDE)

Check list du test d'IHM

- Check all the GUI elements for size, position, width, length, and acceptance of characters or numbers. For instance, you must be able to provide inputs to the input fields.
- Check you can execute the intended functionality of the application using the GUI
- Check Error Messages are displayed correctly
- Check for Clear demarcation of different sections on screen
- Check Font used in an application is readable
- Check the alignment of the text is proper
- Check the Color of the font and warning messages is aesthetically pleasing
- Check that the images have good clarity
- Check that the images are properly aligned
- Check the positioning of GUI elements for different screen resolution.

Conception de votre application

- Partir des besoins du client ou de l'idée du concepteur
- Cahier des charges fonctionnel : cela doit faire quoi ?
- Spécifications :
 - Fonctionnelles : description des fonctionnalités, scénarios, cas d'utilisation
 - Techniques : comment implémenter ces fonctionnalités ? Via quelle architecture ?
- Evaluatez votre travail tout au long du processus et n'hésitez pas à ré-ajuster -> Méthodes Agiles
 - Fonctionnalité par fonctionnalité
 - Avoir toujours une version fonctionnelle pour la tester et ré-ajuster.
 - Prendre en compte les retours utilisateurs

Thématiques de séminaires

- Les outils d'automatisation de test GUI
- Les interfaces du futur