# SUJET D’EXAMEN HM 40 – 22 juin 2021

(Aucun document autorisé)

**Exercice. Questions de cours.**

1. La conception « centrée utilisateur » est un paradigme essentiel dans le développement des IHM. Expliquez en quoi cela consiste et justifier de sa nécessité. Pour aller plus vite, on pourra utiliser un argument ontologique qui place l’utilisateur effectivement au centre du système.
2. Selon le modèle du « processeur humain » la mémoire humaine possède des caractéristiques spécifiques qui la différencient de la mémoire d’un ordinateur du marché. Préciser ces caractéristiques. Peut-on implémenter et simuler ce type fonctionnement dans des applications logicielles. Donner des exemples.
3. La perception est structurée et relève de l’intentionnalité. Par exemple, « que » voyez-vous sur la figure suivante :



Préciser quels principes de la Gestalt sont applicables à cette perception.

1. L’usage des métaphores est très fréquent dans le langage naturel : « cette personne est un glaçon », « il est vert de rage », « le soleil se couche », « ma pelouse a soif ». Il en est de même en conception logicielle et dans le développement des IHM. Citer 4 types de métaphores que vous pouvez rencontrer soit dans les logiciels, soit dans des méthodes de conception ? rappeler domaine source et domaine cible et leur relation. Quel problème peut induire la métaphore ?
2. La notion de fonction comporte des conditions de satisfaction. Quels sont les conditions de satisfaction de la fonction de « portier d’un grand magasin » ? On prendra le cas d’un portier humain. Supposez qu’on remplace le portier humain par un mécanisme de cellule photo-électrique. Quelles sont les conditions de satisfaction de la fonction implémentée ?
3. On peut lire dans certaines méthodes de conception des IHM, des expressions du genre : « une action physique perçue est un acte capté par un dispositif physique d’entrée ». Reprenons la cellule photo-électrique du point précédent, comment interpréter le sens du mot « perception » dans ce cas ? S’agit-il d’une perception réelle du capteur ? quelle conclusion en tirer ? Sinon, expliciter ce dont il s’agit.
4. Le retour d’information en temps réel est essentiel comme critère d’ergonomie et apparait en bonne place dans les recommandations ergonomiques. Essayez de justifier cette règle en se fondant sur la théorie de l’action, en montrant en quoi ce besoin est implicite dans les conditions de satisfaction de l’action, après avoir rappelé en quoi elles consistent.

**Exercice. Structure d’écran**

1) Commentez selon les principes de la structuration d’écran des deux affichages suivants d'un message d'erreur sous deux navigateurs lorsqu'un site WWW n'est pas atteignable. Indiquer les défauts et améliorations entre les deux versions. Donner les diagrammes de structure des écrans et préciser les sujets pragmatiques. Indiquer à l’aide de transitions l’intérêt des modifications apportées aux écrans entre les deux navigateurs.

Navigateur 1 :



Navigateur 2 :



**Exercice**. **Keystroke**.

A votre avis, quel est la méthode la plus rapide pour effacer une partie de texte sous un éditeur standard. Comparons deux techniques envisageables : soit vous vous placez à la fin du texte à supprimer et vous appuyez autant de fois que nécessaire la touche de suppression (*delete*), soit vous sélectionnez l'ensemble du texte indésirable à supprimer à la souris avant d'appuyer sur touche suppression (*delete*). En vous appuyant sur le modèle GOMS-Keystroke, estimez le temps nécessaire pour la suppression d'un texte de *p* caractères adjacents et comparer les deux techniques suivant *p*. Suivez la méthode habituelle : codage, insertion des M, évaluation comparative et numérique. Le résultat est-il conforme à l’expérience ?

Rappel des temps d’exécution :

• Opérateur K : Frappe au clavier ou sélection souris : 200 ms

• Opérateur P : Déplacer la souris sur une cible : 1100 ms

• Opérateur H : Déplacer les mains entre la souris et le clavier : 400 ms

• Opérateur M : Acte de penser à l’opération à effectuer : 1300 ms

**Exercice. Conception.**

Vous devez réaliser un utilitaire interactif de compression/décompression de fichier. Un collègue sera chargé du noyau fonctionnel, soit l’écriture du compresseur/décompresseur, et vous-même de l’interface graphique. Comme son nom l’indique, l’outil permet soit de compresser, soit de décompresser un fichier (unique ici). Il permet notamment de traiter des fichiers très volumineux pour lesquels le temps d’exécution peut prendre des heures.

1. Le collègue vous fournit dans un premier une fonction de type boite noire, conçue selon le principe de modularité, et qui ne retourne que le résultat final. Donner une ligne de commande équivalente au lancement du programme dans une console. Quel problème spécifique cette première approche pose-t-elle au concepteur de l’interface graphique ? En quoi le besoin coté IHM contredit-t-il le principe de modularité ?
2. Le concepteur de l’IHM devra fournir une spécification précise de ces besoins au collègue chargé du noyau fonctionnel. Lors du lancement, une fenêtre graphique s’ouvre et exécute l’application. Il décide d’utiliser SADT d’une part pour l’analyse de tâche et PAC-Amodeous pour représenter la partie logicielle et expliciter le travail demandé au collègue. Ce dernier fournira un ensemble de services permettant de cadencer la compression/décompression à la page, soit en temps suffisamment court, sans plus. Charge au designer de l’interface de compléter le programme.
3. Donner un modèle SADT selon l’approche descendante usuelle. Formaliser la description des entrées/sorties et autres éléments. Préciser l’interface logicielle du compresseur/décompresseur permettant son intégration dans l’IHM.
4. Donner un modèle PAC-Amodeus de la conception logicielle de l’interface. De plus, on supposera l’utilisation d’un langage de programmation de type C++/Qt, disposant des widgets, et des notions de processus (thread) ou de « timer ». Par exemple, comment gérez-vous l’annulation de la tâche, et les autres informations renvoyées à l’utilisateur en temps-réel.
5. Préciser bien les classes et le contrôle, via PAC, et également en combinaison avec un modèle de classes précisant les méthodes et fonctions, avec leur corps pour les plus importantes d’entre-elles.