Modelisation mathematique du Projet MyAssistantTask

1) Definition du problème

Objectif : Développer un assistant virtuel pour aider les utilisateurs à gérer efficacement leurs tâches quotidiennes.

2) Definition des ensembles, contraintes dures et contraintes molles

Ensemble,Variables et paramètres :

T : Ensemble des tâches.

U: Ensemble des comptes utilisateurs.

$P(T)$: Priorité de la tâche T .

$R(T)$: Ensembles des Rappels associés à la tâche T .

$S(U)$: Sécurité des données de l'utilisateur U.

data : Donnée sensibles de l'utilisateur devant être chiffrées

response_time(operation) : Temps de reponse

*availability:*Taux de validité du système

usability_score : Niveau de satisfaction utilisateur

poids(Ti) : poids de la tache Ti

Tâche (Task) : *Ti* où *i* est l'identifiant unique de la tâche.

titre(*Ti*) :Titre

desc(*Ti*) :Description

dateDeb(Ti) : Date de début

dateE(T): Date d'échéance de la tâche T .

hrDeb(Ti) : Heure de début

hrFin(Ti) : Heure de Fin

Status: Status € {en cours, Terminée}

Importance : {0,2,4,6} avec 2 une « importance moderée »; 4 pour une « haute importance » ; 6 pour une « très haute importance », 0 pour « pas important », la valeur de l'Importance est de zero par défaut lors de la création de la tache

Urgence : {0,3,5,7} avec 3 une « urgence moderée »; 5 pour une « très urgente» ; 7 pour une « extrêmement urgente », 0 pour « pas urgente», la valeur de l'Urgence est de zero par défaut lors de la création de la tache

Compte : *Uj* où *j* est l'identifiant unique du compte utilisateur.

nom (Uj): nom de l'utilisateur

numero(*Uj*) : numero de telephone de l'utilisateur

prenom(Uj) : prenom de l'utilisateur

password:(Uj) : mot de passe du compte utilisateur.

mail(Uj) : adresse email de l'utilisateur

photo(Uj) : photo de l'utilisateur

voix(Uj): voix de l'utilisateur

Formulation des fonctions

- ✔ créer une tache :

  Ti ← {titre:titre,desc:desc,dateDeb:dateDeb,dateE:dateE,hrDeb:hrDeb,hrFin:hrFin,status:en cours,Importance:0,Urgence:0}

- ✔ modifier une tache :

  Ti ← TiU{titre:new_titre,desc:new_desc,dateDeb:new_dateDeb,dateE:new_dateE,hrDeb:new_hrDeb, hrFin:new_hrFin}

- ✔ supprimer une tache :

  taskList ← taskList \ {Ti}

- ✔ marquer une tache comme terminé :

  status(Ti) ← terminee

- ✔ Envoyer un rappel :

  sendReminder(Uj,Ti) ← notification(Uj,Ti)

Contraintes dures

- ➢ Les utilisateurs doivent être authentifiés avant d'effectuer toute opération

  ∀*Uj* ∃ *session(Uj) between createTask(Uj,Ti),editTask(Uj,Ti),deleteTask(Uj,Ti).. .n operations in app*

- ➢ Une tâche doit avoir un titre et une date d'échéance définis

$\forall Ti(titre(Ti) \ddagger NULL \wedge dateE(Ti) \ddagger NULL)$

- Toutes les données sensibles doivent être chiffrées

  $\forall data \in \{password, personal\_info\}(data\ est\ chiffré)$

- Une tâche ne peut pas avoir une échéance dans le passé au moment de sa création ou de son édition.

  $dateE(Ti) > current\_time\ v\ new\_dateE > current\_time$

- Une tache ne peut pas être importante et urgente en même temps

  $\neg(important(Ti) \wedge urgent(Ti))$

Contraintes molles

- Le temps de réponse pour une action devrait idéalement être inférieur à 1 seconde.

  $response\_time(operation) < 1seconde$

- L'interface utilisateur devrait être intuitive et facile à utiliser pour 90% des utilisateurs, mesuré par des tests utilisateurs.

  $usability\_score \geq 90\%des\ utilisateurs\ satisfaits$

- Le système devrait être disponible 99.9% du temps

  $availability \geq 99.9\%$

- Les rappels devraient être envoyés au moins 10 minutes avant la date de debut d'une tâche.

  $\forall Ti(rappel\ envoye\acute{} \geq 10minutes\ avantdeadline(Ti))$

Contraintes

- ***Creation de Taches***
  $Ti \leftarrow \{titre:titre, desc:desc, dateDeb:dateDeb, dateE:dateE, hrDeb:hrDeb, hrFin:hrFin, status:en\ cours, Importance:0, Urgence:0\}$

- ***Définition des Rappels :***
  $R(T\,i) = \{rappel\ 1, rappel\ 2, \ldots, rappeln\}\ \forall Ti \in T$

- ***Sécurité des Données :***
  $S(U) = \{cryptage\_donn\ e\acute{}\ es, authentification\_securisee\}\ \forall U \in U$

- ***Importance des Tâches :***
  $I(T\,i) = \{0,2,4,6\}\ \forall Ti \in T$

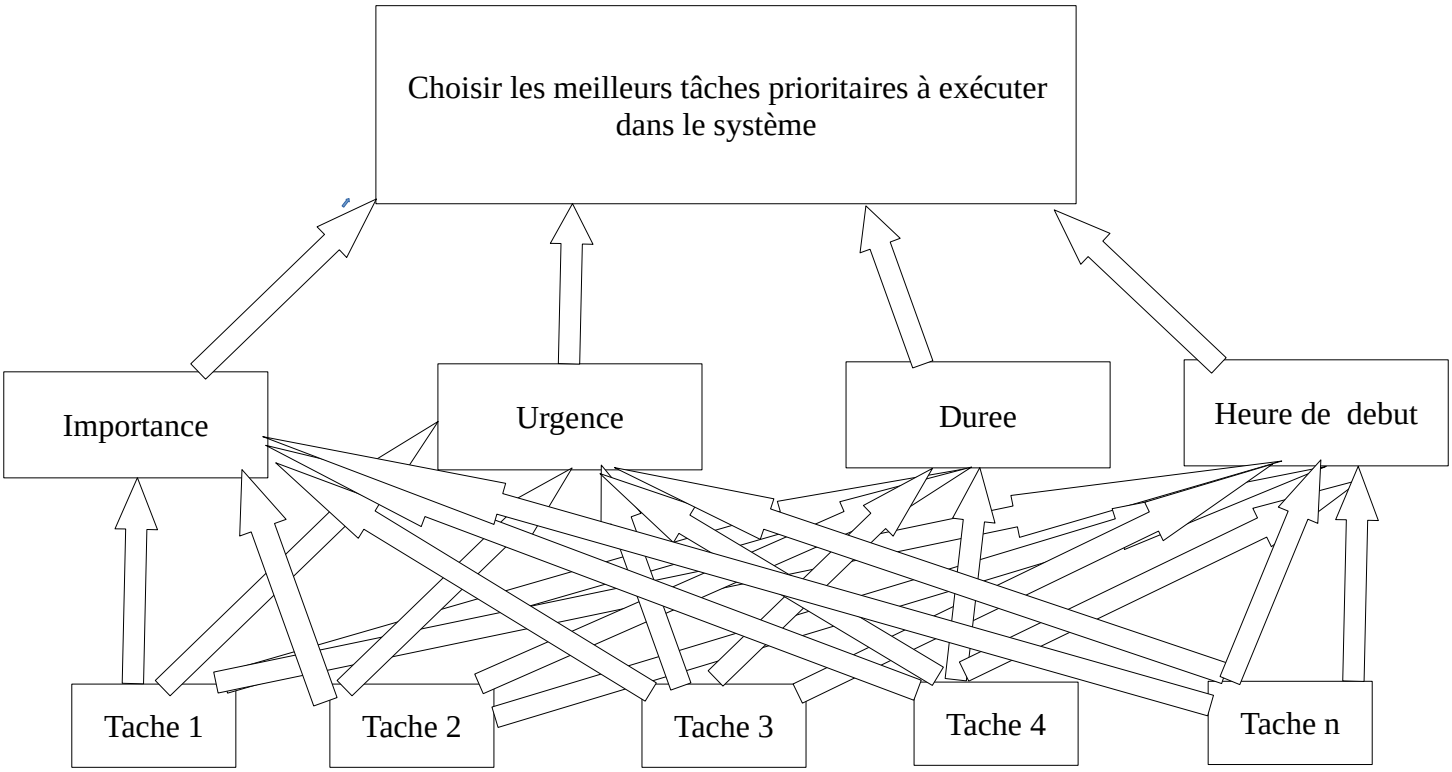
Fonction objective : L'objectif est de minimiser le temps de réponse et de maximiser la satisfaction des utilisateurs.

$Minimiser\ i=1 \sum n\ temps\_d'execution\ des\ taches\ (Ti\ )\ et\ Maximiser\ j=1 \sum m\ satisfaction(Uj\ )$

3- Solution et Techniques AHP

a) Definitions du problème de décision

- **Objectif :** Choisir les meilleurs tâches prioritaires à exécuter dans le système
- **Critères :** Importance, Urgence, Durée, Heure de début
- **Alternatives :** tache1, tache2, la tache3, … tache n



b) Preferences

1- importance egale

3- importance modérée

5- très important

7- vraiment très important

9- extrêmement important

c) Matrice de comparaison par paire

I : importance **U:** urgence **D :** duree (en jours) **H :** heure (en minutes)

	I	U	D	H
I	1	3	5	7
U	1/3	1	3	5
D	1/5	1/3	1	3
H	1/7	1/5	1/3	1
SUM	1,6762	4,5333	9,3333	16,0000

d) Matrice Normalisée

	I	U	D	H	Criteria weight
I	0,5966	0,6618	0,5357	0,4375	0,5579
U	0,1989	0,2206	0,3214	0,3125	0,2633
D	0,1193	0,0735	0,1071	0,1875	0,1219
H	0,0852	0,0441	0,0357	0,0625	0,0569

e) Verification de consistence

Criteria Sum Weight	Lambda
2,3555	4,2222
1,0994	4,1747
0,4919	4,0362
0,2299	4,0408

Lambda max= 4,1185
Consistency Index (CI) = 0,0395
Consistency Ratio (CR) = **0,0439 < 0,1** alors la matrice est consistante

f) Synthèse

Supposons que nous avons quatre taches tache 1,tache 2,tache 3 et tache 4 ayant les valeurs suivantes :

NB : - Importance : {0,2,4,6} avec 2 une « importance moderée »; 4 pour une « haute importance » ; 6 pour une « très haute importance », 0 pour « pas important », la valeur de l'Importance est de zero par défaut lors de la création de la tache

- Urgence : {0,3,5,7} avec 3 une « urgence moderée »; 5 pour une « très urgente» ; 7 pour une « extrêmement urgente », 0 pour « pas urgente», la valeur de l'Urgence est de zero par défaut lors de la création de la tache

- le poids des taches est définit comme suit

poids(Ti) = { 0,3 si l'heure de la tache est comprise entre 1/6 < h<=10h
0,5 si l'heure de la tache est comprise entre 11h < h <= 24h
0,7 si l'heure de la tache est comprise h > 24 h

avec h = hrFin(Ti) – hrDeb(Ti) et D=dateE(*T*i) - dateDeb(Ti)

	I	U	D(en jours)	H(heures)
Tache T1	2	0	5	72
Tache T2	0	3	3	10
Tache T3	4	0	10	10
Tache T4	0	5	1	24

- *par exemple pour le poids de T1, h=72h donc h>24h et son poids sera donc p1=0,7*
- *pour le poids de T2, h=10h donc* 1/6 < h<=10 et son poids sera donc p2=0,3
- *pour le poids de T3, h=10h donc* 1/6 < h<=10 et son poids sera donc p3=0,3
- *pour le poids de T4, h=4h donc* 1/6 < h<=1 et son poids sera donc p4=0,3

En multipliant chaque ligne par le poids correspondant {0,3 ou 0,5 ou 0,7) de la tache en question on obtient:

	I	U	D(en jours)	H(heures)
Tache T1	2*0,7	0*0,7	5*0,7	72*0,7
Tache T2	0*0,3	3*0,3	3*0,3	10*0,3
Tache T3	4*0,3	0*0,3	10*0,3	10*0,3
Tache T4	0*0,3	5*0,3	1*0,3	4*0,3

On aura dont :

	I	U	D	H	Total
Tache T1	1,4	0	3,5	50,4	55,3
Tache T2	0	0,9	0,9	3	4,8
Tache T3	1,2	0	3	3	10,2
Tache T4	0	1,5	0,3	2	3,8

Pour déterminer les meilleurs poids des taches à choisir , nous appliquerons l'algorithme glouton en fonction du total de chaque poids obtenu Total = {55,3;4,8;10,2;3,8}

Algorithme Glouton

1. Définir les Critères de Sélection

Nous avons les critères suivants pour chaque tâche :

- Importance
- Urgence
- Durée
- Heure de début

2. Calculer les Poids Totaux

Les poids totaux pour chaque tâche sont calculés en utilisant l'AHP. :

- T1 : 55,3
- T2 : 4,8
- T3 : 10,2
- T4 : 3,8

3. Classer les Tâches

Classez les tâches en fonction de leur poids total décroissant :

- T1 : 55,3
- T3 : 10,2
- T2 : 4,8
- T4 : 3,8

4. Sélection Gloutonne

Sélectionnons les tâches une par une, en vérifiant les contraintes. **Supposons que nous avons un total de 8 heures de travail disponibles** et que les durées des tâches sont les suivantes :

- T1 : 72 heures
- T3 : 10 heures
- T2 : 10 heures
- T4 : 4 heures

Nous allons sélectionner la tâche avec le poids total le plus élevé qui satisfait les contraintes.

5. Mettre à Jour les Contraintes

I**nitialement, nous avons 24 heures de travail disponibles par jour**. Voici comment nous allons procéder :

(a)Sélection de T1 :

- T1 a le poids le plus élevé (55,3).
- heure de T1 = 72 heures.
- Temps restant après T1 = 24 - 72 = -48 heures.(ne peut pas être accompli dans le temps restant)
- Découper ou Réassigner :
- Diviser T1 en sous-tâches si possible.
- Réassigner à une autre ressource ou négocier une nouvelle échéance.

(b)*Sélection de T3 :*

- T3 a le poids suivant (10.2).
- heure de T3 = 10 heures.
- Temps restant après T3 = **24 -** 10 = 14 heures.
- Tâches sélectionnées : [T3]

(c)Sélection de T2 :

- T2 a le poids suivant (4.8).
- heure de T2 = 10 heures.
- Temps restant après T2 = **14 -** 10 = 4 heures. (car le temps disponible restant etait de 1**4h)**
- Tâche sélectionnée : [T2]

(d)Sélection de T4 :

- T4 a le poids suivant (15).
- heure de T4 = 4 heures.
- Temps restant après T4 = **4 -** 4 = 0 heure. (car le temps disponible restant etait de **4h)**
- Tâches sélectionnées : [T3, T2, T4]

6. Répéter

À chaque itération, nous sélectionnons la tâche avec le poids le plus élevé qui peut être accommodée dans le temps restant. Nous mettons à jour le temps restant et continuons jusqu'à ce que nous n'ayons plus de temps disponible ou plus de tâches pouvant être accommodées.
Dans notre cas, toutes les tâches ont été sélectionnées sans dépasser le temps disponible.

4) Conclusion

Cette modélisation offre un cadre rigoureux pour la gestion des tâches et des utilisateurs dans un système d'assistance virtuelle. En définissant précisément les ensembles, variables, et contraintes, elle assure la sécurité des données, l'efficacité de la gestion des tâches, et une expérience utilisateur optimale. Les contraintes dures garantissent la robustesse, tandis que les contraintes molles visent à améliorer la performance et la satisfaction des utilisateurs. En somme, cette approche fournit une base solide pour développer un système d'assistance virtuelle fiable et performant.