

Série de TD N°2

Concepts fondamentaux

A- Dimensions de l'ingénierie des besoins

1- Dimension WHY

- Pour chaque étude de cas, extraire à partir de l'énoncé le maximum d'objectifs.
- Donner au moins deux alternatives de solutions des problèmes pour chaque étude de cas.
- Déterminer quelques conflits pouvant se présenter dans les trois études de cas.

2- Dimension WHAT

- Donner pour chaque étude de cas, le maximum de contraintes et de suppositions pouvant être extraites de l'énoncé.

3- Dimension WHO

- Déterminer les différentes personnes impliquées dans chaque étude de cas.
- Quelles responsabilités peuvent être définies dans les trois cas ?
-

B- Concepts fondamentaux

1- Besoins du système

Exemple : Toutes les portes doivent être toujours maintenues fermées lorsque le train est en mouvement.

Donner quelques exemples de besoins du système.

2- Besoins du logiciel

Exemple : La variable *doorState* doit toujours avoir la valeur '*closed*' lorsque la variable *measuredSpeed* a une valeur non nulle.

Donner quelques exemples de besoins du logiciel

3- Relations entre les deux types des besoins

Variables de suivi : quantités que le logiciel doit suivre à partir des supports d'entrée (les capteurs)

Variables de commandes : quantités que le logiciel doit contrôler à partir des supports de sortie (actionneurs).

Variables d'entrée : quantités que le logiciel a besoin comme entrées

Variables de sortie : quantités que le logiciel produit pour les sorties.

Définir les besoins systèmes et les besoins logiciel à partir des types de variables ci-dessus.
Donner un exemple illustratif.

Correction Série de TD N°2

Dimensions de l'ingénierie des besoins

4- Dimension WHY

- Pour chaque étude de cas, extraire à partir de l'énoncé le maximum d'objectifs.
Cas1 : accès efficace, minimiser les coûts, unifier les procédures, automatiser, ...
Cas2 : augmenter la capacité de transport, la vitesse et la qualité de service, réduire les gaz polluants, ...
Cas3 : planifier les réunions le plus rapidement possible, satisfaire le max de contraintes, organiser réunion avec succès, minimiser les coûts, ...
- Donner au moins deux alternatives de solutions des problèmes pour chaque étude de cas.
Cas1 : pour minimiser les coûts d'abonnement aux revues : sélection de certaines revues/abonnement aux revues digitales.
Cas2 : pour éviter la collision de deux trains : ne jamais laisser entrer le 2^{ème} train dans un bloc/assurer une distance de sécurité entre deux trains.
Cas3 : pour connaître les contraintes des participants : leurs demander par emails leurs contraintes/accéder directement à leurs agendas électroniques.
- Déterminer quelques conflits pouvant se présenter dans les trois études de cas.
Cas1 : tout le monde est d'accord qu'il faut améliorer l'accès aux livres et revues : procédures de gestion propres à chaque département/unification de ces procédures.
Cas2 : tous sont d'accord qu'il faut assurer un transport rapide : avec ou sans conducteur.
Cas3 : flexibilité des contraintes : participants plus importants/moins importants.

5- Dimension WHAT

- Donner pour chaque étude de cas, le maximum de contraintes et de suppositions pouvant être extraites de l'énoncé.
Cas1 : - fonction ou service à avoir dans le futur système : requête de recherche bibliographique.
- contraintes à satisfaire : couvrir le max de ressources, information pertinente, accès large.
- proposition pour satisfaire l'accès large n'importe où et à n'importe quel moment: accès via internet.

-Suppositions pour cette contrainte : temps de réponse moyen à une requête, interactivité question/réponse pour les non-experts, vie privée (non-staff ne peuvent accéder aux détails des prêts).

6- Dimension WHO

- Déterminer les différentes personnes impliquées dans chaque étude de cas.
- Quelles responsabilités peuvent être définies dans les trois cas ?

Cas1 : objectif : classification efficace, membres du département responsables s'ils ne donnent pas des mots clés efficaces.

Objectif : période de prêt limitée pour accroître disponibilité des livres, prêteurs responsables s'ils ne répondent pas aux rappels, logiciel responsable s'il ne peut générer ces rappels à temps.

Concepts fondamentaux

4- Besoins du système

Exemple : Toutes les portes doivent être toujours maintenues fermées lorsque le train est en mouvement.

Donner quelques exemples de besoins du système.

- *Les enseignants ne peuvent pas emprunter à la fois plus de trois livres.*
- *Les contraintes d'un participant invitées à une réunion devraient être connues le plutôt possible.*

5- Besoins du logiciel

Exemple : La variable *doorState* doit toujours avoir la valeur '*closed*' lorsque la variable *measuredSpeed* a une valeur non nulle.

Donner quelques exemples de besoins du logiciel

- *Le nombre enregistré d'emprunts par un enseignant ne peut jamais dépasser un nombre maximal x .*
- *Une demande de contraintes sera envoyée électroniquement à l'adresse de chaque participant de la liste des invités à la réunion.*

6- Relations entre les deux types des besoins

Variables de suivi : quantités que le logiciel doit suivre à partir des supports d'entrée (les capteurs)

Variables de commandes : quantités que le logiciel doit contrôler à partir des supports de sortie (actionneurs).

Variables d'entrée : quantités que le logiciel a besoin comme entrées

Variables de sortie : quantités que le logiciel produit pour les sorties.

Définir les besoins systèmes et les besoins logiciel à partir des types de variables ci-dessus.

Un besoin système *SysReq* est une relation entre l'ensemble M des variables de suivi et l'ensemble C correspondant des variables de commande. $SysReq \subseteq M \times C$

Un besoin logiciel *SofReq* est une relation entre l'ensemble I des variables d'entrées (inputs) et l'ensemble O correspondant des variables de sorties (outputs). $SofReq \subseteq I \times O$

Donner un exemple illustratif.

