

Modélisation des cas d'utilisation

Par : Abderrazak Sahraoui

Sommaire

- Automatisation de SI
- Analyse des besoins
- Modélisation des besoins par les cas d'utilisation
- Référence pour l'analyse fonctionnelle
- Étude de cas : Guichet Distributeur de billets de banque

Automatisation de Système d'information

- Automatisation de processus
 - Processus de gestion (processus d'entreprise, processus d'affaire)
 - exemple :
 - passer une commande de produit
 - Établir une facture
 - Inscrire un étudiant à un cours
 - Processus Physique (technologie, temps réel)
 - exemple :
 - Régler la température d'une pièce
 - Faire fonctionner un cœur artificiel,
 - Monitorer les système de navigation d'un avion

Définition des besoins

- L'automatisation d'un processus commence par la définition des besoins.
- Cela se fait en explicitant :
 - Le processus en soi
 - Ses objectifs
 - Sa justification
 - Ses activités
 - Les contraintes
 - Législation
 - Réglementation
 - Délais
 - Budget etc...

Définition des besoins (suite ...)

- Les règles
 - Règles résultant de décision de gestion
 - Règles d'usages liés à la profession, à l'environnement etc...
- Les performances exigées
 - Temps de réponse
 - Capacité et volume d'information
 - Technologie (mobilité, sans fils, ..)

Analyse des besoins

- L'analyse des besoins se fait selon
 - différentes perspectives :
 - Les utilisateurs du système à développer
 - Les données (ressources) créées et utilisées dans le futur système
 - Les fonctionnalités attendues du système

Analyse des utilisateurs

- Utilisateur = individu ou autre système qui communique avec le futur système.
- Chercher en quoi chaque utilisateur est dépendant du système
 - Par exemple, il pourrait avoir besoin de rapport
(que se passe-t-il si vous n'obtenez pas le rapport)
- Chercher en quoi le système dépend de chaque utilisateur
 - Par exemple, il pourrait être requis pour remplir des formulaires, approuver des transactions, valider des information, etc...
- Valider la justification de chaque dépendance et vérifier ce qui se passe si cette dépendance n'est pas implémentée

Analyse des utilisateurs (suite ...)

- Chercher les contraintes
 - Y a-t-il des règlements, des lois, des contrats, ... qui imposent la façon de faire ?
- Chercher les règles
 - Y a-t-il des procédures qui déterminent la façon de faire de chaque utilisateur ?
- Chercher les performances requises
 - Combien de personnes pour chaque utilisateur ?
 - Quel temps de réponse ?

Analyse des données

- Il s'agit des données ou ressources manipulées par le système
 - Déterminer les données ou les ressources que chaque utilisateur doit obtenir ou avoir à sa disposition pour effectuer son travail
 - Déterminer les données dont chaque utilisateur est responsable
- Chercher les contraintes
 - Y a-t-il des règlements, des lois, des contrats, ... qui imposent la façon d'acquérir et d'utiliser ces données ?
- Chercher les règles
 - Quel niveau d'autorité est nécessaire pour valider l'acquisition, l'utilisation ou la suppression de chaque donnée (qui est autorisé à utiliser la donnée et qui ne l'est pas) ?
- Chercher les performances requises
 - Combien de volume ?
 - Combien de temps est alloué à chaque transaction?

Analyse des fonctionnalités

- Fonctionnalité : ce que vous attendez que le système fasse
 - Déterminer l'attente:
 - En quoi consiste chaque fonctionnalité et les tâches pour l'accomplir
 - Focaliser sur les objectifs et non sur la façon dont ces objectifs sont atteints.
 - Les gens savent exprimer ce qu'ils font mais ce sont des questions comme :
 - Pourquoi faites-vous ça?
 - Que voulez-vous obtenir?qui leur permettent de rationaliser leurs pensées et les ramènent aux objectifs. C'est la façon pour ne pas se laisser distraire par le processus d'implémentation.
 - D'autres questions à leur poser :
 - Que se passe-t-il si cela ne fonctionne pas ?
 - Y a-t-il plus d'un résultat possible ?
 - Y a-t-il d'autres personnes avec d'autres tâches qui font la même chose que vous ? La font-elles pour la même raison ?

Analyse des fonctionnalités (suite...)

- Chercher les contraintes
 - Y a-t-il des règlements, des lois, des contrats, ... qui imposent la façon de faire ?
- Chercher les règles
 - Quelles sont les procédures qui définissent la façon de faire ?
 - Y a-t-il des dépendances entre les tâches ?
- Chercher les performances requises
 - Dans quel temps maximal la tâche doit-elle être accomplie?
 - Quels sont les facteurs qui influencent ou déterminent le temps nécessaire?

Analyse des fonctionnalités (suite...)

- Pièges à éviter
 - Ne pas faire de suppositions :
 - Chaque information non validée est suspecte
 - Confirmer tout
 - Confirmer de plusieurs façons
 - Ne pas dupliquer l'implémentation existante
 - L'analyse devrait mener toujours à questionner le status-quo
 - Ne pas confondre préférences et besoins
 - Utiliser des techniques de modélisation qui permettent de regarder de manière objective les problèmes et identifier les incohérences

Les cas d'utilisation

- Les utilisateurs doivent définir leurs objectifs et leurs besoins vis-à-vis du futur système.
- Les cas d'utilisation servent à structurer ces objectifs et ces besoins.
- Chaque cas d'utilisation doit illustrer une image de la fonctionnalité du système. On précise pour cette fonctionnalité, les utilisateurs concernés, ce qu'ils doivent fournir comme information, ce qu'ils obtiendront comme résultat et les tâches qu'ils doivent effectuer pour obtenir ce résultat.

Modèle de cas d'utilisation

- Le modèle des cas d'utilisation d'un système est constitué de:
 - diagramme UML des cas d'utilisation illustrant QUI devrait pouvoir faire QUOI, grâce au système ;
 - diagrammes d'activité ;
 - descriptions narratives pour chaque cas.

Constituants d'un cas d'utilisation

- La définition d'un cas d'utilisation se fait par :
 - le nom du cas d'utilisation
 - La description sommaire du cas
 - Les acteurs interagissant avec le système
 - Un acteur est une entité externe qui agit sur le système .
 - Le terme acteur ne désigne pas seulement les utilisateurs humains mais également les autres systèmes.
 - Les acteurs sont des classificateurs qui représentent des rôles et non pas des personnes physiques. Ce sont des acteurs types.
 - Les préconditions
 - Si les préconditions ne sont pas remplies l'acteur se fera refuser l'utilisation du cas
 - La description narrative des actions constituant :
 - Le scénario nominal d'utilisation
 - Les scénarios alternatifs
 - Les cas d'exception ou erreurs
 - Les postconditions
 - Conditions qui doivent être vérifiées (garanties à la fin du cas d'utilisation)

Structuration du diagramme des cas d'utilisation

- Après avoir identifié les cas d'utilisation principaux, il est utile de structurer l'ensemble des cas d'utilisation en mettant en évidence :
 - les cas partagés
 - les cas optionnels
 - les cas similaires ou de la même famille
- UML offre trois types de relations permettant de mettre en évidence ces cas:
 - relation d'inclusion : formalisée par la dépendance « includes »
 - pour rendre compte des cas partagés
 - relation d'extension : formalisée par la dépendance « extends »
 - pour rendre compte des cas optionnels
 - relation de généralisation / spécialisation
 - pour rendre compte des cas de la même famille

Relation d'inclusion entre Cas

- Lors de la description des actions de cas d'utilisation, il apparaît qu'il existe des sous-ensembles communs à plusieurs cas d'utilisation, il convient donc de factoriser ces fonctionnalités en créant de nouveaux cas d'utilisation et les reliés par une dépendance de type « includes » avec les cas d'utilisation qui les avaient en commun.
- Un cas d'utilisation A utilise un cas d'utilisation B signifie qu'un lien de type « includes » part de A vers B.

Relation d'extension entre Cas

- La relation <<extends>> montre une possibilité d'exécution d'interactions qui augmenteront les fonctionnalités du cas étendu, mais **de façon optionnelle**, non obligatoire, alors que la relation <<includes>> suppose une **obligation** d'exécution des interactions.
- On utilise principalement cette relation pour séparer le comportement optionnel (les variantes) du comportement obligatoire.
- *Le cas d'utilisation B étend le cas d'utilisation A ; cela signifie que A utilisera B si une certaine condition est vérifiée.*

Relation de généralisation entre Cas

- Cette relation est à prendre au sens classique de la *spécialisation*, inhérent à l'héritage.
- De façon générale en objet, une situation de cas particulier se traduit par une relation de généralisation.
- **On peut également généraliser les acteurs**

Package (Paquetage) de cas d'utilisation

- La notion de « package » permet de mettre en œuvre un partitionnement des modèles en ensembles cohérents tout en indiquant les dépendances entre les différents packages.

Pour construire un diagramme de Cas:

- lire attentivement le cahier des charges ;
- bien comprendre le système à étudier ;
- délimiter le système à étudier et fixer des frontières au problème ;
- retrouver les acteurs qui interagissent avec le système ;
- organiser les acteurs par relation de généralisation spécialisation si c'est pertinent ;
- pour chaque acteur, rechercher les cas d'utilisation du système c-à-d les fonctionnalités attendues ;
- le système au complet peut être modélisé par un diagramme de cas d'utilisation représentant :
 - les acteurs du système
 - les cas d'utilisation (fonctionnalités)
 - les liens entre acteurs et cas d'utilisation
 - quel acteur accède à quel cas d'utilisation ?

Pour construire un diagramme de Cas: (suite...)

- compléter les cas d'utilisation principaux par des cas d'utilisation internes. Les cas d'utilisation internes sont des précisions d'autres cas d'utilisation ;
- les cas d'utilisation internes sont reliés au cas initiaux par des relations stéréotypées de type « includes » ou « extends », ou par une relation de spécialisation ;
- la description narrative d'un cas d'utilisation se fait par des scénarios textuels qui définissent la suite logique des actions qui constituent ce cas :
 - Il est possible de définir des scénarios simples ou des scénarios plus détaillés faisant intervenir les variantes, les cas d'erreurs, etc.
- les autres requis ou besoins non-fonctionnels peuvent être décrits textuellement dans une section qui peut être ajoutée aux descriptions des cas d'utilisations.

Référence pour l'analyse fonctionnelle

<http://www.fonctionnel.net/profession/analyste-profil-qualites>

Étude de Cas 1: Guichet Automatique de Banque (GAB)

- Tout utilisateur doit s'authentifier avec sa carte pour pouvoir utiliser les services du GAB. L'authentification de la carte se fait à même la carte.
- Les clients de la banque peuvent consulter leur solde, retirer de l'argent, déposer des chèques ou de l'argent comptant. Le GAB est connecté au système de la banque pour accéder aux informations des clients.
- Les transactions de retrait des clients de la banque sont validées par le système d'information de la banque;
- Les clients d'autres banques affiliée au système Interac peuvent seulement retirer de l'argent;
- Les transactions de retrait pour les usagers qui ne sont pas des clients de la banque, sont validées par le système Interac.

Étude de Cas 1: Guichet Automatique de Banque (suite)

- Un opérateur de maintenance passe recharger le distributeur à chaque matin et récupère les chèques déposés et les cartes avalées lorsque le client enregistre trois échecs d'affilé dans l'authentification d'une carte.