Ce document récapitule l'exercice de conception architecturale avec ADD réalisé en LOG430-01, automne 2015, cours 12.

CE DOCUMENT N'EST PAS DESTINÉ À ÊTRE DISTRIBUÉ AUX ÉTUDIANTS. Il s'agit d'un document visant à structurer l'exercice pour les enseignants. L'exercice a été fait au tableau de façon interactive en LOG430-01, session A15.

Pour les besoins de cet exercice, un document séparé de quatre pages a été fourni aux étudiants. Il s'agit essentiellement d'une traduction des spécifications du guichet automatique bancaire (GAB) utilisé au laboratoire 1, augmenté d'une restructuration des scénarios de qualité que le groupe a lui-même produit plus tôt cette session pour un GAB.

L'exercice a aussi été réalisé avec ADD v1, mais avec des ajouts issus de ADD v3, qui est sur le point d'apparaître dans un nouveau livre en voie de publication. Toutefois, les étapes telles que numérotées dans ADD v1 ont été conservées.

# Avant de commencer l'exercice

ADD requiert en entrée les Architectural Drivers, soit le sous-ensemble des exigences qui a un impact important sur l'architecture. Ce sous-ensemble est identifié de façon plus ou moins subjective.

Nous avons d'abord priorisé les scénarios de qualité avec les deux priorités, comme dans une ATAM (importance du scénario pour le succès du système, et difficulté anticipée à accommoder le scénario dans l'architecture). Ceci a été fait informellement de façon interactive avec tout le groupe. Ça a généré des discussions intéressantes. ADD v3 propose aussi de lier chaque scénario de qualité à tous les cas d'utilisation pertinents. Ceci est éventuellement utile pour "allouer" les scénarios de qualité aux éléments architecturaux. Voici le résultat pour le groupe LOG430-01 A15. Les scénarios surlignés ont été retenus comme Architectural Drivers:

|  |  |  |  |
| --- | --- | --- | --- |
| ID scénario | Importance pour succès | Difficulté à accommoder | Cas d'utilisation pertinents |
| D1 | L | L | Tous |
| D2 | H | L | CU3-4-5-6-7-8, ext.1 |
| D3 | H | H | CU3-4-5-6-7-8, ext.1 |
| D4 | H | L | CU3-4-5 |
| D5 | n/a[[1]](#footnote-1) | n/a | CU3-4-5-6-7-8, ext1 |
| M1 | L | L | CU3-4 |
| M2 | L | L | CU3-4 |
| P1 | M | L | CU3-4-5-6-7-8, ext1 |
| P2 | M | L | CU3-4 |
| S1 | H | L | CU3-4-5, ext1 |
| T1 | H | L | CU3-4-5-6-7-8, ext1 |
| U1 | M | L | CU3-4-5-6-7-8 |
| Port1 | H | TBD[[2]](#footnote-2) | CU1-2-3-4-5-6-7-8, ext1 |

Pour les cas d'utilisation, il a été décidé collectivement d'y aller au plus simple pour cette première itération. CU3 et CU4 sont incontournables, car à la base de tout. Certains argumentaient qu'on pourrait s'en tenir à ça. D'autres argumentaient que ce serait bien de concevoir pour au moins une fonction concrète parmi les quatre. Comment choisir? Proposition: ce qui est le plus critique pour le succès. Autre critère: ce qui sollicite un maximum d'éléments du système. Autre critère: ce qui est le plus risqué (plusieurs étudiants souhaitaient commencer par ce qui est le plus facile, priorité "L" dans colonne "difficulté à accommoder", ce qui va à l'encontre d'une bonne gestion du risque). Le transfert semblait au départ un bon candidat (2 comptes à spécifier, 2 opérations), mais ce choix sollicite un minimum d'éléments périphériques. Le dépôt et le retrait sont égaux par rapport à cet aspect, le retrait a été retenu comme étant l'opération la plus courante. Donc CU3-4-5 sont les "Architectural Drivers" fonctionnels.

Enfin, on a d'abord choisi CT3-4 parmi les contraintes, mais on s'est vite rendu compte qu'il fallait aussi considérer CT2.

Il est suggéré de saisir comme suit les "architectural drivers" de départ:

|  |  |
| --- | --- |
| Exigences fonctionnelles | * CU3 - session * CU4 - transaction * CU5 – retrait |
| Exigences d'attributs de qualité | * D2 – Processus GAB complet en panne * D3 – connexion réseau GAB-banque * T1 – tests unitaires transactions * Port1 – connectivité ISO 8583 |
| Contraintes | * CT2 – API CEN XFS * CT3 – Communication TCP/IP sur Ethernet * CT4 – Connectivité Interac |
| Préoccupations (développement d'un nouveau système dans un domaine mature) | * Proposer une structure initiale du système * Proposer des structures pour les fonctions principales * Proposer des structures pour les qualités principales |

# ADD, itération 1

| Étape | Détails |
| --- | --- |
|  | L'élément architectural est tout le GAB. il est recommandé de démarrer avec un diagramme de contexte. Le GAB est interconnecté à deux systèmes externes, soit le réseau Interac et la banque. Pas clair que le premier n'inclut pas le second, TBD. Diagramme omis ici par manque de ressources. |
| 2a. | Tous les "architectural drivers" ci-haut sont pertinents pour cette itération:   * CU3 - session * CU4 - transaction * CU5 – retrait * D2 – Processus GAB complet en panne * D3 – connexion réseau GAB-banque * T1 – tests unitaires transactions * Port1 – connectivité ISO 8583 * CT2 – API CEN XFS * CT3 – Communication TCP/IP sur Ethernet * CT4 – Connectivité Interac |
| 2b. | Il est recommandé ici d'essayer de trouver une architecture de référence. De telles architectures existent typiquement dans des domaines maitrisés (applications web, Rich Internet Applications, applications mobiles, etc). Pour un GAB, nous n'avons rien trouvé, et avons dû identifier des styles et tactiques pertinents. Une pause d'apprentissage avec directive d'identifier des tactiques candidates et des patrons candidats a donné ce qui suit:   |  |  | | --- | --- | | Décision, localisation | Exposé des motifs | | Tactique heartbeat du GAB vers la centrale. Une exigence spécifie 15s de délai maximal de détection de panne, nous avons fixé le heartbeat à une période de 10 s pour garder de la marge. | D2 - Pour détecter la perte du GAB à partir de la centrale en moins de 15s | | Ping/echo du GAB vers la centrale, fréquence TBD. | D3 – Pour détecter dans le GAB la perte de connexion entre le GAB et la centrale | | Utilisation du style architectural déploiement, en se concentrant d'abord sur le matériel (vue physique) | Afin de raisonner à propos des connexions entre le "cœur" du guichet et tous ses périphériques | | Utilisation du style architectural client-serveur | * Afin de gérer les interactions entre le GAB et la centrale * Afin de gérer les interactions entre le cœur du GAB et ses périphériques | |
| 2b-c. | Ébauches de vues architecturales, allocation des exigences.  **Vue physique[[3]](#footnote-3) du GAB**: diagramme en étoile avec le "cœur du GAB" au centre, et les divers périphériques et systèmes externe en étoile autour:   * un lecteur magnétique de carte-client; * une console client (clavier et écran) pour l'interaction avec le client; * une fente pour recevoir les enveloppes de dépôt; * un distributeur pour fournir de l'argent (par multiples de 20 $); * une imprimante pour émettre des reçus imprimés; * un interrupteur à clé pour permettre à un opérateur de démarrer ou d'arrêter la machine. * l'imprimante pour les logs * la connexion avec le réseau Interac * la connexion avec la banque   Tableau des éléments de la vue physique   |  |  | | --- | --- | | Élément | Description | | Cœur du GAB | Contient toute la logique d'interaction avec le client, avec la banque, avec le réseau Interac, avec tous les périphériques, et enfin toutes les fonctions principales. | | Lecteur de carte magnétique | Dispositif qui reçoit la carte magnétique de chaque client. | | Console (écran) | Écran tactile qui affiche les informations aux clients, contient parfois des "boutons" sur la périphérie (choix des comptes, etc) | | Clavier | Permet aux clients de saisir les informations (NIP, montants) | | Récepteur enveloppes dépôts | Permettre de recevoir des enveloppes de dépôts contenant de l'argent (billets seulement) et/ou des chèques | | Distributeur billets | Fournir les liquidités aux clients lors de retraits, en multiples de $20. | | Imprimante | Émettre des reçus imprimés aux clients | | Interrupteur à clé | Permettre à un opérateur de démarrer ou d'arrêter la machine. | | Imprimante logs | Imprime toutes les transactions. Cette imprimante se trouve dans la partie blindée du guichet avec l'argent et les enveloppes, car la journalisation des transactions est une information hautement sensible. | | Réseau vers banque | Représente le matériel de connexion vers la banque | | Réseau Interac | Représente le matériel de connexion vers le réseau Interac |   La nature des liens de communications entre le cœur du GAB et les autres éléments reste à déterminer (ports USB, Ethernet, autre,…).  **Vue modules interne du cœur du GAB**: décomposition modulaire. L'indentation dans la liste suivante indique les niveaux de décomposition   * Module "Cœur du gab"   + Module "Session"   + Module "Transaction"   + Module "Retrait" (hérite du module "Transaction")   + Module "Heartbeat"   + Module "Ping"   Tableau des éléments de la vue modules "coeur du GAB"   | Élément | Responsabilités | | --- | --- | | Cœur du GAB | Toutes les responsabilités indiquées plus haut sur l'élément du même nom dans la vue physique. | | Session | Module responsable d'implémenter le CU3 – session   * interaction avec lecteur cartes; * interaction avec écran (messages) * interaction avec clavier (NIP) * affichage du menu – choix de transaction * gestion des transactions * gestion de la demande du NIP | | Transaction | Module responsable d'implémenter le CU4 – transaction   * transaction instanciée par session * interaction avec les périphériques communs à toutes les transactions:   + banque (validation)   + clavier (montants, comptes)   + écran (messages)   + imprimante reçus clients   + imprimante logs | | Retrait | Module responsable d'implémenter le CU5 – retrait   * interaction avec les périphériques spécifiques à ce type de transaction: distributeur de billets | | Heartbeat | Module responsable d'mettre le Heartbeat pour signaler la disponibilité du AGB à la centrale. | | Ping | Module responsable de "pinger" la centrale et gérer les actions en cas de perte de connexion avec la centrale | |
| 2d. | Pas fait. |
| 2e. | Vérification: la présentation suivante est dorénavant recommandée. Tant que tous les AD de départ ne sont pas dans la colonne "Complètement fait", on sait qu'il reste des itérations à faire.   |  |  |  |  | | --- | --- | --- | --- | | Pas fait | Partiellement fait | Complètement fait | Décisions architecturales dans cette itération | |  | CU3 |  | Allocation physique, vue modulaire haut niveau avec module dédié, style client-serveur | |  | CU4 |  | Allocation physique, vue modulaire haut niveau avec module dédié, style client-serveur | |  | CU5 |  | Allocation physique, vue modulaire haut niveau avec module dédié, style client-serveur | |  | D2 |  | Tactique "Heartbeat", allocation à un module dédié | |  | D3 |  | Tactique "Ping/echo", allocation du "Ping" à un module dédié. Le mécanisme de "failover" (mise des transactions en cache) reste à faire, ne semble pas évident. | |  | T1 |  | Allocation des fonctionnalités à des module dédiés (Transaction, Retrait) | |  | Port1 |  | Allocation physique. Il reste à établir ce que l'utilisation de la norme ISO 8583 implique. | | CT2 |  |  | Aucune. | |  | CT3 |  | Allocation physique. | |  | CT4 |  | Allocation physique. | |

Il s'agirait ensuite de choisir un autre élément à décomposer parmi ceux obtenus à la première itération, et à reprendre les mêmes étapes.

1. Le serveur est hors portée dans cette itération de conception. [↑](#footnote-ref-1)
2. ISO 8583 ne nous est pas connue. Considéré risqué, ce scénario a été retenu. Il faudrait analyser la norme ASAP dans la vraie vie. [↑](#footnote-ref-2)
3. Une représentation de la distribution du matériel semblait appropriée comme première structure ici, afin de canaliser les considérations convenablement. [↑](#footnote-ref-3)