

Méthodes de développement

Guide pour la rédaction d'un document de conception préliminaire

Sommaire

1 - Objet du guide.....	2
2 - Plan type.....	2
3 - Rédaction du document.....	2
3.1 Introduction	2
3.1.1 Objet	2
3.1.2 Terminologie et sigles utilisés.....	2
3.1.3 Documents de référence	2
3.2 Présentation générale de l'architecture	2
3.3 Les principaux objets et leurs classes.....	3
3.3.1 Liste des principales classes	3
3.3.2 Relations entre classes.....	3
3.4 Scénarios d'interaction entre objets	3
3.5 Les processus et le déploiement	3
3.6. Bases de données.....	4
3.7 Dimensionnements et performances	4
4 - Exemple de document d'architecture logicielle	4
Exemple de document d'architecture.....	5

1 - Objet du guide

Ce guide fournit un plan type de document de conception préliminaire dans le cadre de méthodes agiles.

Un exemple de document partiellement rédigé basé sur l'exemple présenté dans l'introduction aux guides de rédaction est joint.

2 - Plan type

Le plan type suivant est recommandé pour un document réduit dans le cadre d'une conception objet :

1	Introduction 1.1 Objet 1.2 Terminologie et sigles utilisés 1.3 Documents de référence
2	Présentation générale de l'architecture
3	Les principaux objets et leurs classes 3.1 Liste des principales classes 3.2 Relations entre classes
4	Scénarios d'interaction entre objets
5	Les processus et le déploiement
6	Bases de données
7	Dimensionnements et performances

3 - Rédaction du document

3.1 Introduction

3.1.1 Objet

Préciser l'objet du document (description de l'architecture), le nom du système et brièvement sa fonction.

3.1.2 Terminologie et sigles utilisés

Définir les abréviations employées et les termes qui ont un sens particulier.

3.1.3 Documents de référence

Donner les références des documents ayant servi de référence pour l'architecture (la SGB principalement) ou pouvant aider à comprendre le problème posé.

3.2 Présentation générale de l'architecture

Présenter rapidement l'organisation du logiciel en indiquant les principaux éléments de l'architecture et les principaux choix faits, par exemple :

- découpage en programmes, répartition du logiciel entre les calculateurs,

- processus, et threads éventuels,
- architecture réseau et accès réseau,
- accès aux bases de données (par serveur base de données, base réseau...),
- conception objet,
- mode de programmation (événementiel ou séquentiel...),
- langages et systèmes d'exploitation,
- principaux logiciels réutilisés...

3.3 Les principaux objets et leurs classes

3.3.1 Liste des principales classes

Les principales classes pourront être présentées par catégorie sous forme de tableaux indiquant :

- le nom de la classe,
- l'indication du rôle principal de la classe.

3.3.2 Relations entre classes

Les relations entre les principales classes identifiées ci-dessus seront représentées sur des diagrammes de classes montrant les classes et les diverses relations entre classes (généralisation, association...).

Pour que les diagrammes soient lisibles, il faudra répartir les classes sur plusieurs diagrammes. On pourra par exemple répartir les principales classes en packages de classes, et faire pour chaque package de classes un diagramme montrant les classes du package, et les classes des autres packages en relation avec ces classes.

3.4 Scénarios d'interaction entre objets

Les scénarios d'interaction seront définis à partir des cas d'utilisation. On choisira les séquences d'événements les plus significatives et on construira un diagramme d'interaction entre objets (diagramme de collaboration ou diagramme de séquence de messages) pour chaque séquence retenue.

3.5 Les processus et le déploiement

Présenter le découpage du logiciel en programmes et les processus attachés aux programmes. Pour les processus multi-threads, indiquez le rôle de chaque thread. Les présentations pourront être faites sous forme de schémas ou de tableaux.

Quelques explications sur le choix des découpages pourront être fournies.

Le déploiement sera représenté sous forme d'un schéma montrant les principaux matériels, les réseaux et les interfaces externes, avec indication des processus exécutés sur chaque calculateur.

On pourra compléter cette partie si cela s'avère utile par les principes de fonctionnement des processus (sous forme de diagrammes) et les interactions entre processus ou threads (diagrammes de séquence de messages).

3.6. Bases de données

Lorsque le programme utilise des bases de données, on précisera :

- le contenu des bases de données,
- l'organisation des données (par exemple un schéma conceptuel de données),

3.7 Dimensionnements et performances

On précisera les dimensionnements ou contraintes de performances significatifs pour l'architecture, par exemple :

- nombre maximum d'utilisateurs ou nombre d'utilisateurs simultanés,
- tailles maxima de tables,
- temps de réponse
- délais à respecter...

Pour chacun de ces points, on précisera comment l'architecture les prend en compte.

4 - Exemple de document d'architecture logicielle

La suite de ce document est un document d'architecture logiciel réduit partiellement rédigé. Il répond au démonstrateur de navettes d'aéroport présenté dans l'introduction aux guides de rédaction.

On notera les points suivants :

1 - Comme tout document d'un projet, un dossier d'architecture commence par une page d'entête et une page historique, ces deux pages n'ont pas été reproduites ici.

2 - Le modèle de classe présenté n'a fait l'objet d'aucune validation, il est montré uniquement pour présenter des exemples de rédaction ou de présentation, et pas pour servir de modèle d'architecture

3- Les anglo-saxons ne connaissant pas les accents, tous les identificateurs destinés à la programmation ont été mis sans accents.

4 - Les principales classes résultent :

- du problème physique : Borne, Ligne, Station, Navette
- des structures de données et l'accès à la base de données : DataMngr, DataListe, DataFric, DataDate, DataUser,
- de la gestion des interfaces réseau : Rezo, Formateur, Serveur,
- de la gestion d'autres interfaces : Paiement, Login, Autorisation,
- des bases d'architecture : MainAT, ModelAT, FormAT, GestionAT, MainCC, ModelCC, FormCC, GestionCC.

5 - Pour les diagrammes de classe, il a été décidé de faire 5 diagrammes, chacun contenant un groupe de classes principales (qui apparaît comme un package dans les diagrammes) et les classes des autres groupes en relation avec ces classes.

Exemple de document d'architecture

Sommaire

1 - Introduction.....	6
1.1 Objet	6
1.2 Terminologie et sigles utilisés.....	6
1.3 Documents de référence	6
2 - Présentation générale de l'architecture.....	6
3 - Les principaux objets et leurs classes	7
3.1 Liste des principales classes	7
3.2 Diagrammes de relations entre classes	8
4 - Scénarios d'interaction entre objets.....	9
5 - Les processus et le déploiement.....	9
6 - Base de données	10
6.1 Contenu des bases de données	10
6.2 Schéma des bases de données	11
7 - Dimensionnements et performances	11

1 - Introduction

1.1 Objet

Ce document décrit l'architecture du démonstrateur de navettes d'aéroport, dont le rôle est de fournir des informations sur des bornes à disposition des clients d'autobus navettes pour aéroport.

1.2 Terminologie et sigles utilisés

Ligne

Trajet suivi par un autobus entre l'aéroport et un terminus et inversement, avec des arrêts intermédiaires

Station

Endroits où l'autobus est arrêté ou s'arrête pour faire monter ou descendre des passagers : aéroports, terminus de lignes, arrêts intermédiaire de ligne.

...

1.3 Documents de référence

Spécification générale de besoin du démonstrateur de navettes d'aéroport N° du

2 - Présentation générale de l'architecture

Le logiciel est découpé en deux programmes :

- un programme PAT pour la borne,
- un programme PCC pour les fonctions centre de contrôle.

Sur chaque borne, un processus s'exécute sur le programme PAT, il créera des threads pour les échanges réseau.

Au centre de contrôle, un processus s'exécute sur le programme PCC et crée deux threads pour les échanges réseau avec chaque borne active.

Les échanges entre les bornes et le centre de contrôle sont fait par l'intermédiaire d'un réseau (fourniture extérieure au système) utilisant les protocoles TCP-IP et ayant une vitesse de transmission limitée pour certains sites d'implantation de bornes.

Afin de limiter les échanges par le réseau, on plantera une base de données sur chaque borne et au centre de contrôle. Les bases de données de chaque borne sont identiques et sont un sous ensemble de la base de données du centre de contrôle. Les accès aux bases de données seront faites avec ODBC et la programmation sera prévue indépendante du SGBD. Pour le centre de contrôle on pourra utiliser des SGBD comme Oracle ou SQL Server. Pour les bornes on utilisera de préférence un SGBD qui gère la base dans un fichier (Access, SQLite...) afin de simplifier les rechargements de base.

La conception des programmes est de type objet, et le mode de programmation événementiel. Les programmes sont codés en C++ pour Windows XP avec la bibliothèque QT.

3 - Les principaux objets et leurs classes

3.1 Liste des principales classes

Données partagées (Data)

Classe	Rôle principal
Ligne	Informations sur une ligne
Station	Informations sur un arrêt
Navette	Informations sur une circulation de navette
Borne	Informations sur une borne

Base de données (DB)

Classe	Rôle principal
DataMngr	Connexion à la base et exécution des requêtes
DataLigne	Accès aux données lignes, stations, horaires
DataFric	Accès aux tarifs et cumuls de ventes
DataDate	Date, heure et saint du jour
DataUser	Accès aux données opérateur (login, droits, mots de passe)

Classes réseau (Net)

Classe	Rôle principal
Rezo	Accès au réseau (borne et centre de contrôle)
Serveur	Gestion du réseau (centre de contrôle)
Formateur	Formatage et déformatage pour les échanges réseau

Classes particulières à la borne (AT)

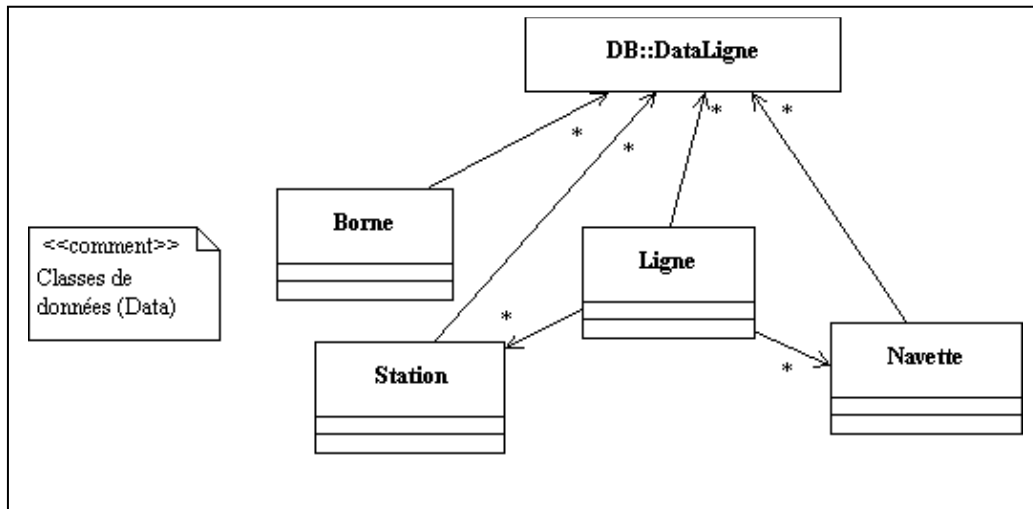
Classe	Rôle principal
MainAT	Gestion de la fenêtre principale et construction de l'application
FormAT	Gestion de la fenêtre Vue principale
GestionAT	Gestion de la borne
ModelAT	Accès aux données partagées
Paielement	paiement des billets

Classes particulières au centre de contrôle (CC)

Classe	Rôle principal
MainCC	Gestion de la fenêtre principale et construction de l'application
FormCC	Gestion de la fenêtre Vue principale
GestionCC	Gestion du centre de contrôle
ModelCC	Accès aux données partagées
Login	Ecran de login
Autorisation	Gestion des autorisations carte bancaire

3.2 Diagrammes de relations entre classes

Classes de données partagées (Data)



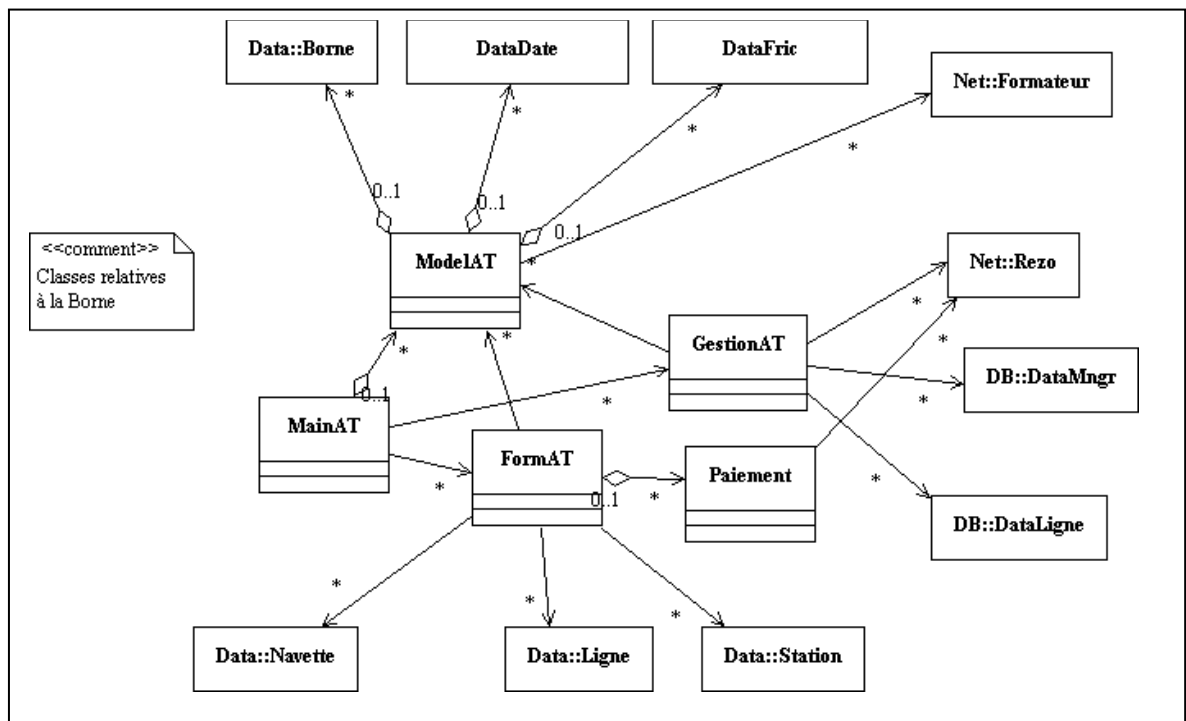
Classes Base de données (DB)

...

Classes réseau (net)

...

Classes relatives à la borne



Classes relatives au centre de contrôle

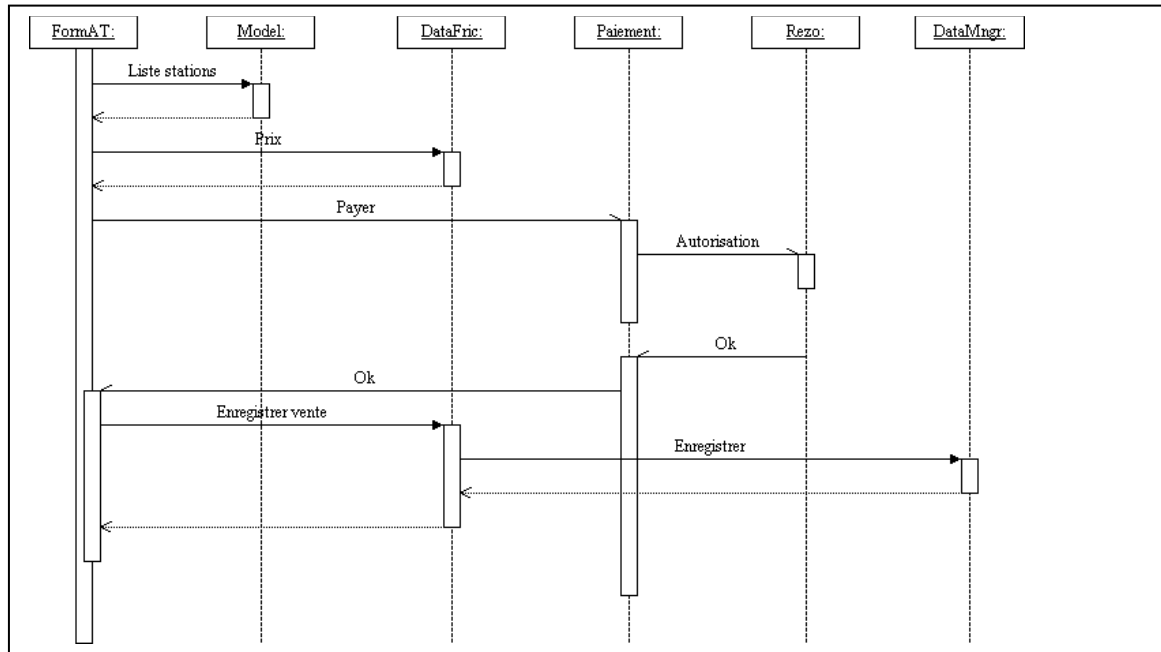
...

4 - Scénarios d'interaction entre objets

Affichage des prochains départs

...

Vente de billets



Réception de message réseau

...

Initialisation de la borne

...

Initialisation du centre de contrôle

...

Surveillance des bornes au centre de contrôle

...

5 - Les processus et le déploiement

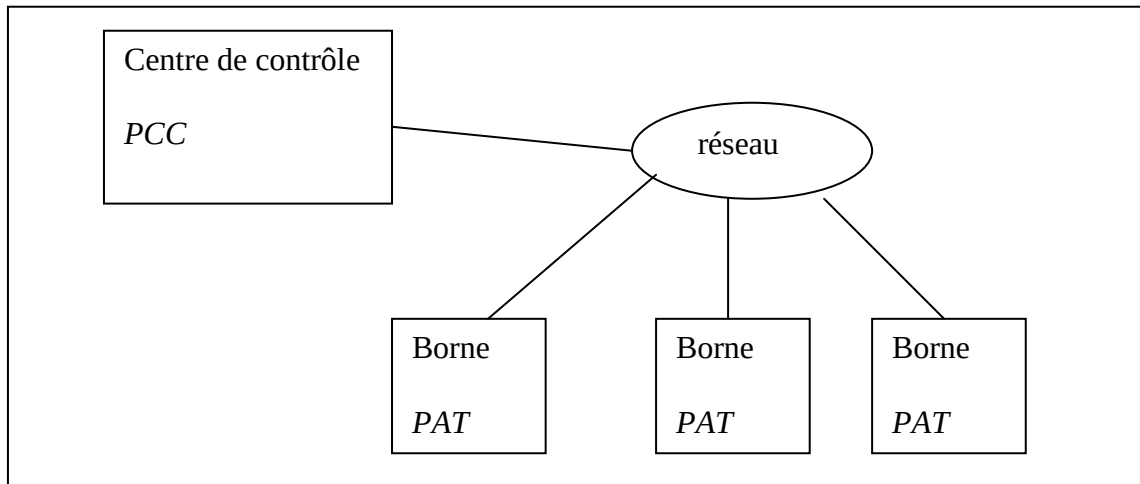
Programmes et processus

Processeur	Program me	Processus	Rôle
Borne	PAT	1 processus PAT	Gestion d'une borne
Centre de contrôle	PCC	1 processus PCC	Opérations du centre de contrôle

Le processus PAT crée deux threads pour les échanges réseau.

Le processus PCC crée deux threads pour les échanges réseau avec chaque borne active.

Le déploiement du système peut être représenté par le schéma suivant (les processus sont indiqués en italique) :



Nota

Le réseau est une fourniture extérieure au système.

6 - Base de données

6.1 Contenu des bases de données

Les données nécessaires pour le fonctionnement du système sont stockées dans des bases de données au centre de contrôle et dans chaque borne.

La base de données comprend :

- la table des lignes donnant pour chaque ligne le code, le sens et le terminus,
- la table des stations donnant pour chaque station son nom, sa localisation et sa zone tarifaire,
- la table ligne - stations fournissant les associations ligne - station -station suivante,
- la table des navettes fournissant les associations lignes - navettes,
- la table des horaires fournissant les associations navette - station - heure de passage
- la table des tarifs indiquant pour chaque tarif la période de validité et le prix pour chaque zone tarifaire,
- la table des bornes indiquant pour chaque borne le nom, la station d'implantation et son état,
- la table calendrier indiquant le saint du jour pour chaque jour de l'année
- la table des messages textuels, qui enregistre pour chaque message émis, la date et heure d'émission, l'émetteur, les destinataires et le contenu du message,
- la table des cumuls qui enregistre borne par borne pour chaque jour le nombre de billets vendus par zone tarifaire et type de carte de paiement, et les montants correspondants,
- la table des mots de passe qui contient les identificateurs et mots de passe (sous forme codée) des opérateurs et administrateurs du centre de contrôle.

Pour dimensionner ces tables, on retient les valeurs suivantes :

Nombre de lignes possibles	10
Nombre maximum d'arrêts par ligne	10
Nombre maximum de zones tarifaires	10

Nombre maximum de circulations journalières sur une ligne	10 dans chaque sens
Nombre maximum de bornes	50
Nom de station, d'opérateur, d'administrateur	20 caractères maximum
Code navette	4 chiffres
Taille maxima d'un message textuel	40 caractères
Taille des mots de passe	Entre 8 et 16 caractères

Les utilitaires SGDB seront utilisés pour la création de la base, les sauvegardes et restauration, ainsi que pour les opérations de type suppression de lignes, de stations, de messages...

6.2 Schéma des bases de données

...

7 - Dimensionnements et performances

Les paramètres dimensionnants pour les tables de la base de données sont indiqués au § 6.1. Ces paramètres seront pris en compte dans la définition détaillée et la création de la base de données.

Le centre de contrôle doit pouvoir assurer les dialogues avec 5 bornes simulées parmi 50 bornes possibles. Les tables seront dimensionnées pour au moins les 50 bornes possibles, et les écrans seront définis pour représenter les informations d'au moins 5 bornes.

Les temps de réponse à une demande sont limités à 2 secondes. Des essais seront fait sur les premiers prototypes pour vérifier que ce délai est bien tenu pour les requêtes courantes, et voir quelles sont les types de requête qui risquent d'être plus longues et demanderont l'affichage d'un sablier.