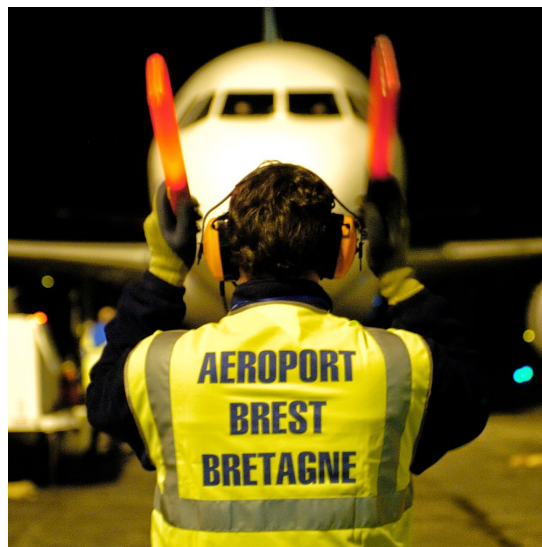


RAPPORT DE PROJET SIMULATION

UV 5.8 Simulation

Simulation de l'aéroport de Brest



Rédigé par :

Alice Danckaers
Thomas Boulter

Sous la direction de :

Olivier Verron



ENSTA
Bretagne

Option Systèmes Perception Information Décision

© 2016 Alice Danckaers and Thomas Boulier.

Licensed under the Creative Commons Attribution-ShareAlike 4.0 International Public License.

Sommaire

1	Objectifs de l'étude	4
2	Analyse du problème	5
3	Modélisation du système	6
4	Implémentation du modèle	7
5	Compte-rendu de V & V	9
6	Résultats de la simulation	10
7	Analyse des résultats	11
8	Perspectives d'évolutions	12
	Annexes	12
	Index	13
	Glossaire	13

Chapitre 1

Objectifs de l'étude

L'objectif de la présente étude est de simuler l'aéroport de Brest afin de dimensionner de façon optimale ses futures évolutions.

Pour cela nous avons réalisé une simulation à événement discret, dans laquelle nous avons étudié l'état de l'aéroport à chaque événement.

L'aéroport possède un seul terminal avec 4, 6 ou 8 emplacements ; une seule piste (6L/24R) et deux taxiways entre l'aérogare et la piste, un pour l'atterrissage et l'autre pour le décollage. Cette configuration est représentée sur la figure 1.1.

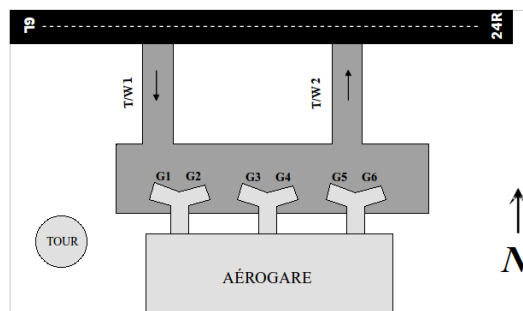


FIGURE 1.1 – Boucle de simulation

La simulation porte une attention particulière sur les arrivées des avions et les temps qu'ils mettent à effectuer les différentes étapes, de leur arrivée dans l'espace aérien de l'aéroport jusqu'à leur départ. La simulation a pour objectifs de déterminer les surcharges éventuelles de l'aéroport et leur contexte, en fonction des infrastructures disponibles. Elle est donc modulable, afin de faire varier ces infrastructures, notamment en modifiant les nombres de portes d'accès et de pistes, et d'étudier les conséquences de ces variations.

Le livrable final est composé du programme de simulation (sous la forme d'un projet JAVA compilable) et d'une étude des données générées par ce programme. Cette étude déterminera notamment les moyennes de fréquentation journalière, de retards, de temps d'attentes avant atterrissage.

Chapitre 2

Analyse du problème

Le système que nous allons simuler est un aéroport. Toutefois, notre modèle peut-être grandement simplifié par rapport à un aéroport réaliste. Puisque l'étude se focalise sur la surcharge du terminal, seul le processus d'atterrissage, de décollage et de traitement des avions nous intéresse. On peut donc ignorer de nombreux aspects, tels que la gestion des personnels, l'entretien des infrastructures, la gestion financière. La seule chose pertinente dans notre cadre est le status des avions et des infrastructures (occupés ou non) au cours du temps. c'est à partir de ces informations que nous pourrions extraire les statistiques de fréquentation demandées.

Chapitre 3

Modélisation du système

Le système que nous simulons est composé de divers objets.

Les entités représentent les acteurs de la simulation. Cela inclut les lieux. L'aéroport représente l'ensemble de la simulation. Il contient une piste, deux taxiways, plusieurs portes d'embarquement. Il reçoit des avions, qui sont générés en grand nombre au cours de la simulation.

Les variables de la simulation sont moins nombreuses : La date permet de tenir compte du temps dans notre simulation. La météo (qui est soit bonne soit mauvaise) modifie le temps d'approche.

Les événements sont le cœur de la simulation. Ils sont déclenchés séquentiellement par la boucle de simulation. Ils sont tous datés. Ils planifient d'autres événements, modifient les entités et les variables et renvoient les informations qui serviront à l'analyse des résultats. L'événement de notification crée l'avion et le met en attente si la piste et le taxiway 1 ne sont pas disponibles. L'événement d'approche rend indisponible la piste et le taxiway 1. Il dure 2 à 5 minutes, doublé en cas de mauvais temps. L'événement d'atterrissage dure 2 minutes. Dès qu'il est fini, l'événement de relachement de piste rend celle-ci disponible. L'événement de roulage entrant occupe le taxiway 1 et dure 2 à 6 minutes. Dès qu'il est terminé, le taxiway 1 est libéré par l'événement correspondant si une porte est libre. Les événements de déchargement, de ravitaillement et d'embarquement durent respectivement 10, 30 et 20 minutes. À l'issue de ceux-ci, si le taxiway 2 est libre, l'événement de roulage sortant est déclenché et l'avion libère la porte pour occuper le taxiway 2. Le roulage dure là encore 2 à 6 minutes. Si la piste est libre, l'événement de décollage est déclenché. Il dure 3 minutes, à l'issue desquelles la piste devient à nouveau libre.

L'événement de changement de météo existe en parallèle des autres, il agit sur la variable de météo pour la passer de bonne à mauvaise et inversement.

Chapitre 4

Implémentation du modèle

Le système est composé de deux parties : un programme, écrit en Java, qui implémente la boucle de simulation, et un fichier Excel qui contient les données issues du programme et qui les traite pour en ressortir des résultats statistiques. Le programme Java implémente une boucle de simulation événementielle, telle que décrite par la figure 4.1, dans la classe principale Aéroport. Le processus d'initialisation remplit un agenda (de type `<Events>SortedList`) qui contient les événements d'arrivées d'avion dans l'espace aérien de l'aéroport, ainsi que les changements de météo. Ces événements sont générés par une boucle. Les intervalles de temps entre les événements dépendent de la date de l'événement précédent. Ceci est fait pour respecter le cahier des charges. Nous avons fait le choix de générer tous les événements d'arrivée d'avion et de changement de météo avant le parcours de la boucle principale de simulation. L'initialisation instancie également les objets représentant les infrastructures (tous de type `Facilities`), les avions étant générés durant la simulation par le traitement de l'événement correspondant à leur arrivée. Tous les événements sont des sous-classes de la classe abstraite `Events`. Chacun contient sa propre méthode *doSomething()* qui est appelée par la boucle de simulation. Cette méthode crée les événements dont découle l'événement courant et renvoie une chaîne de caractère décrivant l'événement qui sera intégrée dans le logger, le fichier retourné par le programme. L'architecture du programme Java est décrite dans le diagramme de classe 4.2.

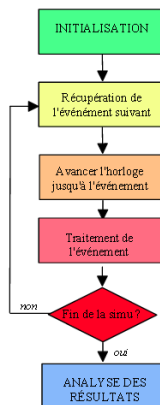


FIGURE 4.1 – Boucle de simulation

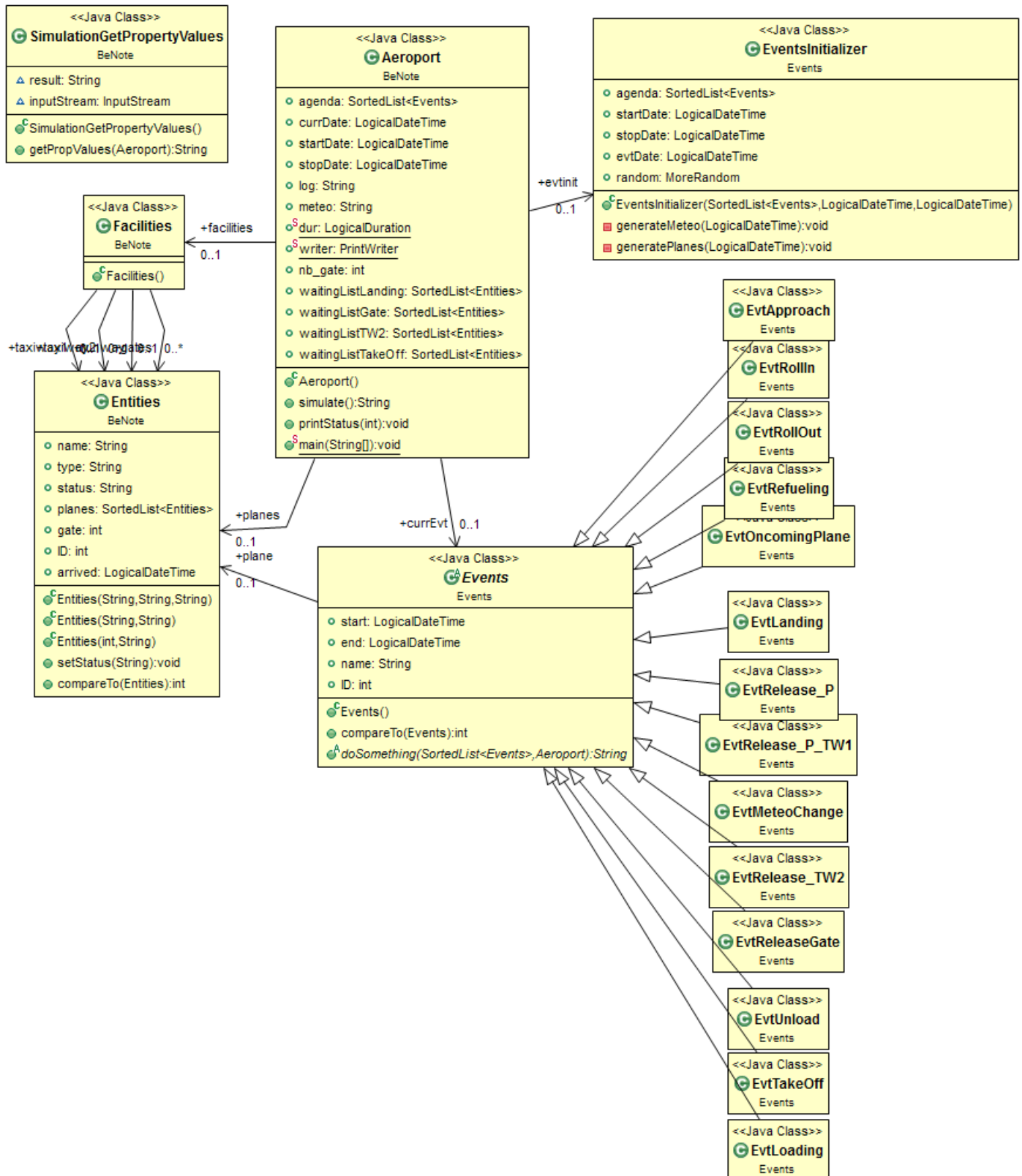


FIGURE 4.2 – Diagramme de classe

Chapitre 5

Compte-rendu de V & V

Revue critique du fonctionnement et des résultats.

Chapitre 6

Résultats de la simulation

Synthèse des résultats. On utilisera des graphiques (histogrammes, camemberts, etc.) pour faciliter la lecture.

Les résultats seront fournis sous forme de fichiers texte au format CSV ou .

Chapitre 7

Analyse des résultats

Commentaire des résultats et réponses apportées au problème posé.

Chapitre 8

Perspectives d'évolutions

Suggestions d'amélioration du logiciel de simulation, du modèle...

