Rapport de projet simulation

UV 5.8 Simulation

Simulation de l'aéroport de Brest

Rédigé par :

Alice Danckaers
Thomas Boulier

Sous la direction de :

Olivier Verron

ENSTA Bretagne

Option Systèmes Perception Information Décision

Sommaire

1 Objectifs de l'étude **4**

2 Analyse du problème **5**
2.1 L'aéroport 6
2.2 Les avions 7
2.3 Les infrastructures 9
2.4 Le scénario 10

3 Modélisation du système **11**

4 Implémentation du modèle **13**

5 Compte-rendu de V & V **15**

6 Résultats de la simulation **16**
6.1 Retard en fonction du nombre de portes 16
6.1.1 Aéroport à 4 portes d'embarquement 17
6.1.2 Aéroport à 6 portes d'embarquement 17
6.1.3 Aéroport à 8 portes d'embarquement 18
6.1.4 Aéroport à 12 portes d'embarquement 19
6.2 Répartition journalière des arrivées 20

7 Analyse des résultats **22**
7.1 Un aéroport saturé 22
7.2 Des solutions au problème 23

8 Perspectives d'évolutions **24**
8.1 Amélioration de la simulation 24
8.2 Amélioration du modèle 24

9 Conclusion **25**

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.

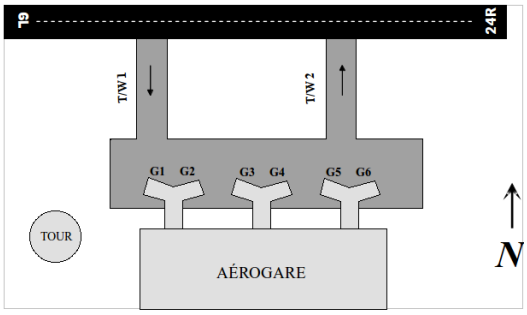

FIGURE 1.1 – Boucle de simulation

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 statut des avions et des infrastructures (occupés ou non) au cours du temps. c'est à partir de ces informations que nous pourront extraire les statistiques de fréquentation demandées. Ainsi les systèmes environnant en interaction avec le système étudié sont essentiellement constitué des Avions.

Les entités de simulations identifiées à la lecture du sujet sont :

- L'aéroport de Brest
- Les avions
- les infrastructures de l'aéroport, a savoir :
    - la piste
    - un taxiway utilisé à l'atterrissage
    - plusieurs portes (4, 6 ou 8)
    - un taxiway utilisé au décollage

Le cycle de vie des infrastructures est bien sur inclus dans celui de l'aéroport. C'est l'aéroport qui donne créé ses infrastructures. Nous allons dans les paragraphes qui suivent analyser chaque entité et identifier puis caractériser :

- leurs variables d'états
- leurs propriétaires d'initialisation et caractéristiques
- les événements temporels, et d'états
- les comportements et hypothèses de modélisation

2.1 L'aéroport

Entité	Aéroport
Type d'Entité	Permanente
Variables d'états	meteo currDate waitingListLanding waitingListGate waitingListTW2 waitingListTakeOff
Variables statistiques de scrutation	
Paramètres techniques et données d'initialisation	startDate stopDate evtinit nb_gate
Événements	Aucun
Processus Stochastique	Aucun

TABLE 2.1 – Analyse de l'entité aéroport

Les données d'initialisation de l'aéroport sont les suivantes :

- La date d'ouverture de l'aéroport (début de la simulation)
- La date de fermeture de l'aéroport (fin de la simulation)
- La météo au début de la simulation
- Le nombre de portes

Elles sont lues dans le fichier de configuration (config.properties) et permettent de configurer l'état initial de l'aéroport.

Les états de l'aéroport sont :

- Les conditions météorologique au niveau de l'aéroport, c'est-à-dire la météo
- La date de l'événement courant
- Une liste d'attente qui contiendra les avions qui ont contactés la tour de contrôle mais qui doivent attendre la libération de la piste et du taxiway 1 pour atterrir.
- Une liste d'attente qui pourra contenir un avion qui attendra qu'une porte se libère sur le taxiway 1
- Une liste d'attente qui contiendra les avions qui attendent à une porte la libération du taxiway 2 pour repartir

– Une liste d'attente qui pourra contenir un avion qui attendra que la piste se libère sur le taxiway 2 pour décoller.

Il n'y a pas d'événements temporels qui sont directement liés avec l'aéroport, car on a mis en place l'entité infrastructure

2.2 Les avions

Entité	Avion
Type d'Entité	Temporaire
Variables d'états	status
Variables statistiques de scrutation	Aucune
Paramètres techniques et données d'initialisation	ID type
Événements	EvtOncomingPlane EvtApproach EvtLanding EvtRollIn EvtUnload EvtRefueling EvtLoading EvtRollOut EvtTakeOff
Processus Stochastique	Process de durée de roulement à l'arrivée Process de durée d'approche pour l'atterrissage Process de durée de roulement au départ

TABLE 2.2 – Analyse de l'entité avion

Les variables d'initialisation portent sur :

– le numéro d'identification ID

– le type d'entité. Il a la même valeur pour tous les avions. Il permet d'identifier cette entité comme étant bien un avion

Les états de l'avion sont :

– En attente de libération de la piste et du taxiway pour atterrir

– En approche

– En cours d'atterrissage

- En roulement sur le taxiway 1
- En attente sur le taxiway 1
- En cours de déchargement
- En cours de réapprovisionnement
- En cours de chargement
- En attente de libération du taxiway 2
- En roulement sur le taxiway 2
- En attente sur le taxiway 2
- En cours de décollage

Les événements temporels identifiés sont :

- L'avion contacte la tour pour signifier son approche
- L'avion réalise la phase d'approche de la piste
- Atterrissage de l'avion
- Roulement de l'avion en entrée sur le taxiway 1
- Le déchargement de l'avion
- Le ravitaillement de l'avion
- Le chargement de l'avion
- Le roulement de l'avion en sortie sur le taxiway 2
- Le décollage de l'avion

La durée de certains événements est déterminée par un processus stochastique, conformément à ce qui a été demandé dans le cahier des charges. Le roulement en sortie et en entrée prends entre 2 et 6 minutes. La durée d'approche pour l'atterrissage prends entre 2 et 5 minutes par beau temps et le double dans le cas contraire.

2.3 Les infrastructures

Entité	infrastructures
Type d'Entité	Permanente
Variables d'états	runway.status taxiway1.status taxiway2.status gates[k].status (k de 0 à nb_gate)
Variables statistiques de scrutation	Aucune
Paramètres techniques et données d'initialisation	Aucun
Événements	EvtRelease_P_TW1 EvtReleaseGate EvtRelease_TW2 EvtRelease_P
Comportement	Aucun
Processus Stochastique	Aucun

TABLE 2.3 – Analyse de l'entité infrastructure

Il n'y a pas de paramètres d'initialisation des infrastructures. Elles sont toujours composées d'une piste, de deux taxiway et d'une liste de portes dont le nombre est défini à la création de l'aéroport.

Les états des infrastructures sont accessibles via le champs status des entités filles qui la composent.

Les événements temporels identifiés sont :

- La libération de la piste et du taxiway 1
- La libération d'une porte
- La libération du taxiway 2
- La libération de la piste seule

2.4 Le scénario

Entité	agenda
Type d'Entité	Permanente
Variables d'états	Aucune
Variables statistiques de scrutation	offset
Paramètres techniques et données d'initialisation	startDate stopDate
Événements	EventsInitializer
Comportement	generatePlane generateMeteo
Processus Stochastique	Process de génération de l'arrivée des avions Process de génération de la météo

TABLE 2.4 – Analyse de l'entité aéroport

Les variables d'initialisation portent sur :

- La date de début de simulation
- La date de fin de simulation

Le seul événement temporel identifié est l'événement d'initialisation du scénario qui va générer les changements de météo ainsi que les arrivées d'avions. Ceci est effectué par un processus stochastique.

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 infrastructures.

L'aéroport représente le moteur 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. On peut ainsi voir le séquencement de certains événements ci-après.

- L'événement de contact de la tour 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 et cette durée est doublée en cas de mauvais temps.
- L'événement d'atterrissage dure 2 minutes. Dès qu'il est fini, l'événement de libération de la 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'issu 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 ficher 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'avions dans l'espace aérien de l'aéroport, ainsi que les changement de météo. Ces événement 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 tout les événements d'arrivées d'avions et de changement de météo avant le parcours de la boucle principale de simulation. L'initialisation instancie également les objets représentant les infrastructures (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.

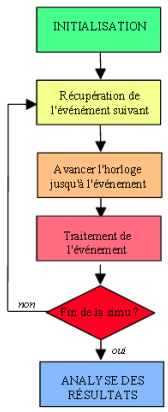

FIGURE 4.1 – Boucle de simulation

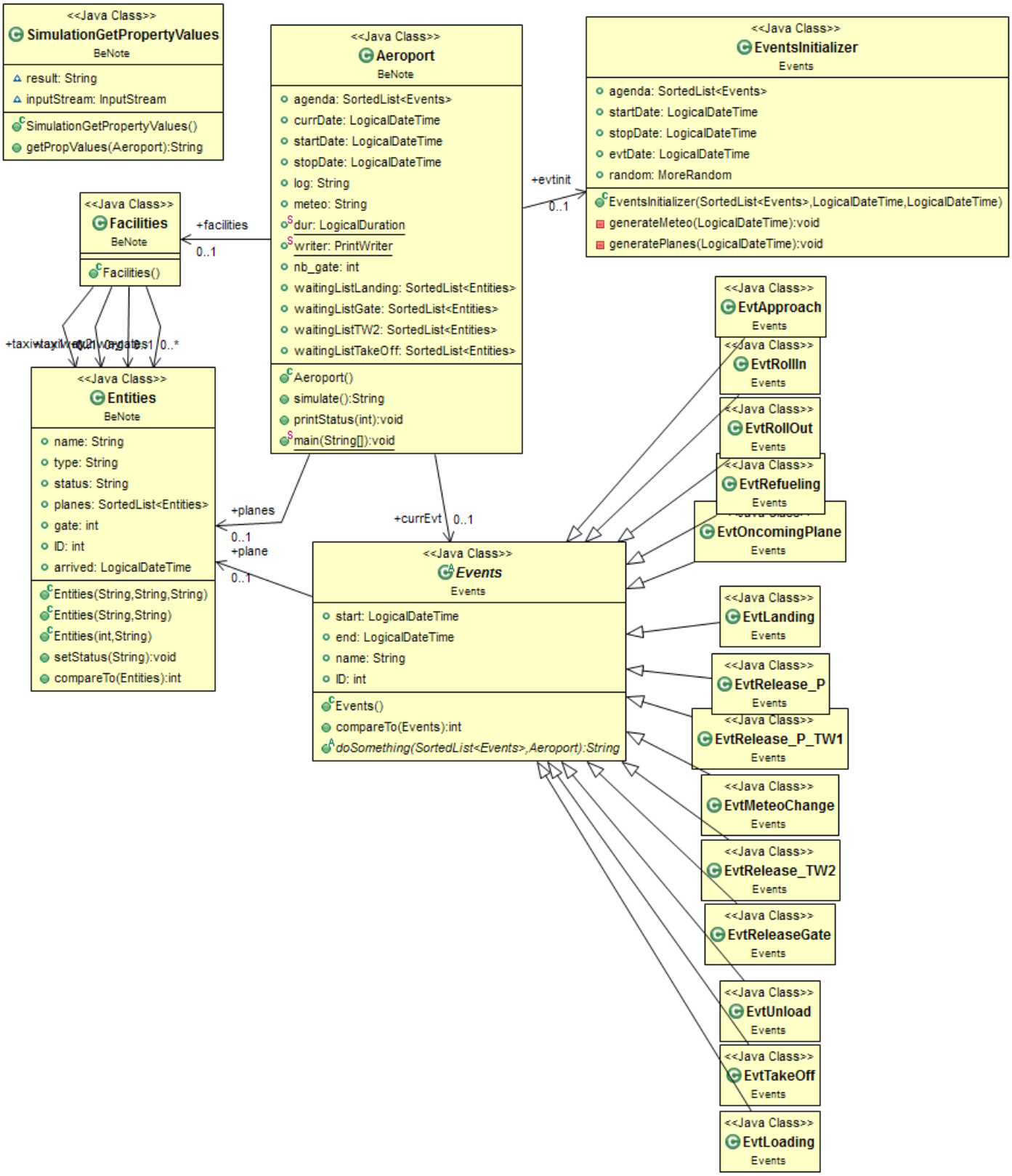

FIGURE 4.2 – Diagramme de classe

Chapitre 5

Compte-rendu de V & V

Le suivi de la qualité du code a été effectuée tout au long du développement. En premier à l'aide des fonctionnalités d'Éclipse, qui permettent à chaque instant de vérifier les dépendances entre classes, packages et le typage des entrées-sorties. Ensuite à l'aide de la fonction *print_ status()*, qui affiche les états des infrastructures dans la console et peut être utilisée par exemple après chaque événement. De plus, la première version des événements affichaient les log dans la console, plutôt que de les enregistrer, ce qui a permis des tests de fonctionnement. Enfin, même après la fin de la phase de développement, nous avons gardé un esprit critique sur nos résultats, ce qui a permis notamment de revenir sur une incohérence dans la libération de la piste lors de la phase d'atterrissage : l'avion ne libérait la piste qu'à la fin de la phase de roulage, ce qui empêchait un éventuel avion en attente de décoller durant ce laps de temps.

Chapitre 6

Résultats de la simulation

La simulation a été exécutée sur 4 scénarios. Chaque scénario durait 90 jours, et voyait des avions arriver avec des probabilités identiques :

- Un avion toutes les 20 minutes en moyenne en horaires normaux.
- Un avion toutes les 10 minutes en heure de pointe : de 7h à 10h et de 17h à 19h, en semaine.
- Un avion toutes les 40 minutes le week-end.
- Aucun avion de 22h à 7h.

Tous les avions ont le même comportement, tel qu'il est décrit dans la modélisation du problème (voir chapitre 3).

Les quatre scénarios ont pour seule différence le nombre de portes d'embarquement instanciées : 4, 6, 8 ou 12.

On étudie les horaires de notification des avions à la tour de contrôle et de début de phase d'approche. La différence entre ces deux dates indique le retard de l'avion, c'est-à-dire le temps que l'avion a passé en l'air à attendre la permission d'atterrir. Les moyennes de ces retards sont compilées dans le tableau 6.1

4 portes	6 portes	8 portes	12 portes
123 :12 :35	544 :06 :03	356 :29 :14	0 :08 :10

TABLE 6.1 – Retard moyen

6.1 Retard en fonction du nombre de portes

Voici ci dessous des répartitions plus détaillées des retards en fonction du nombre de portes.

6.1.1 Aéroport à 4 portes d'embarquement

La répartition des retards pour un aéroport à 4 portes,en pourcentage d'avions dans des intervalles de retards est donnée en figure 6.1.

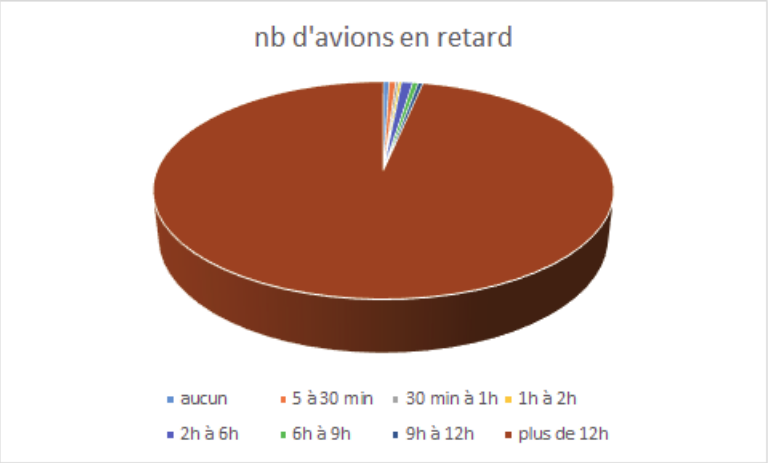

FIGURE 6.1 – Répartition des retards, avec 4 portes

Une autre donnée intéressante est la répartition des retards par jour, donnée par la figure 6.2.

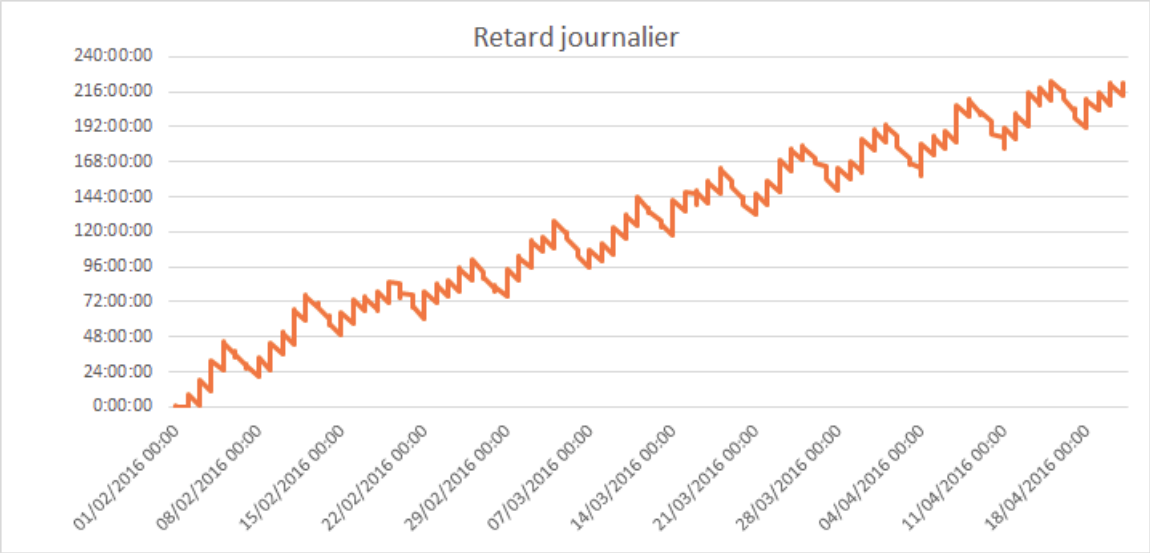

FIGURE 6.2 – Retard avec 4 portes

6.1.2 Aéroport à 6 portes d'embarquement

La répartition des retards pour un aéroport à 6 portes,en pourcentage d'avions dans des intervalles de retards est donnée en figure 6.3.

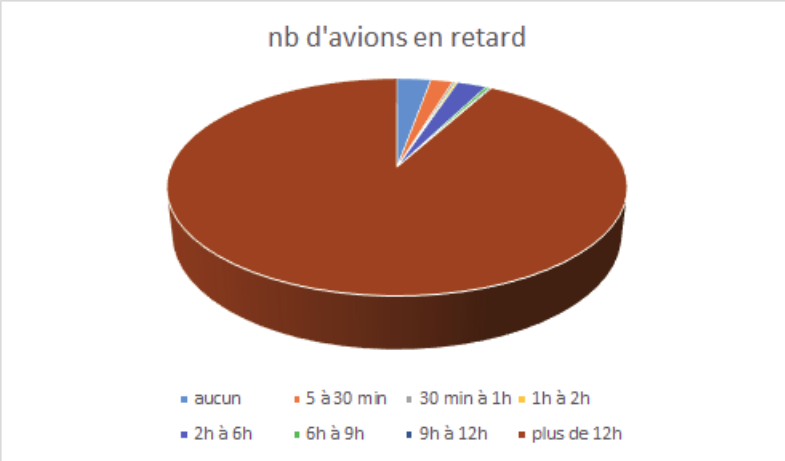

FIGURE 6.3 – Répartition des retards, avec 6 portes

De même que précédemment la répartition des retards par jour est donnée par la figure 6.4.

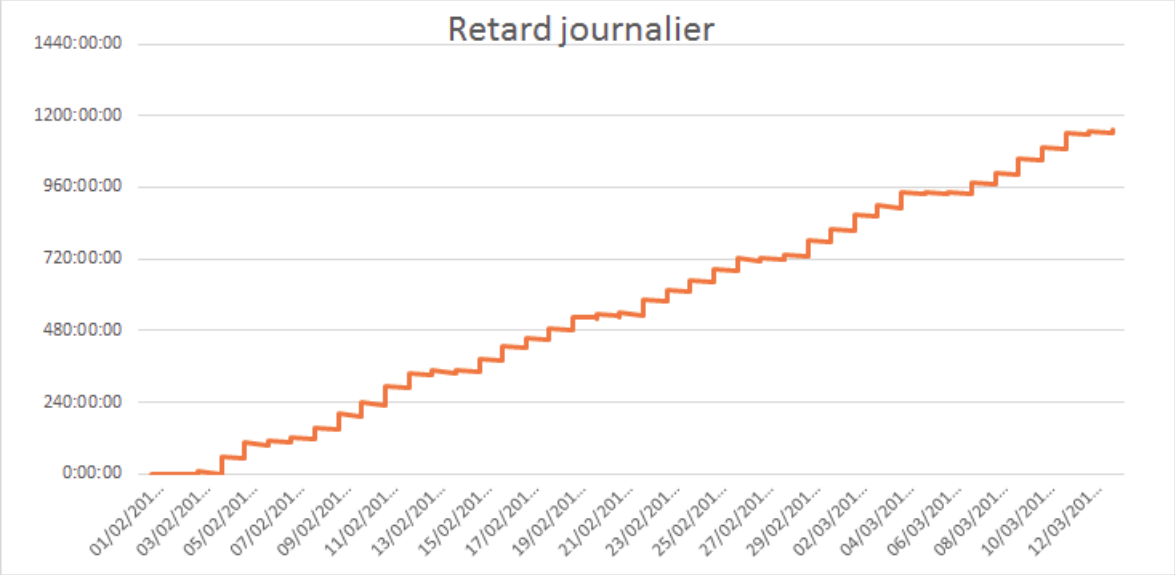

FIGURE 6.4 – Retard avec 6 portes

6.1.3 Aéroport à 8 portes d'embarquement

La répartition des retards pour un aéroport à 8 portes,en pourcentage d'avions dans des intervalles de retards est donnée en figure 6.5.

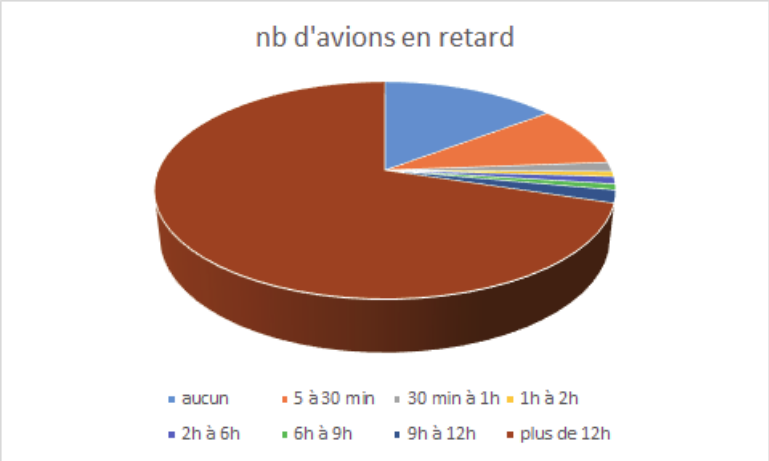

FIGURE 6.5 – Répartition des retards, avec 8 portes

La répartition des retards par jour est donnée par la figure 6.6.

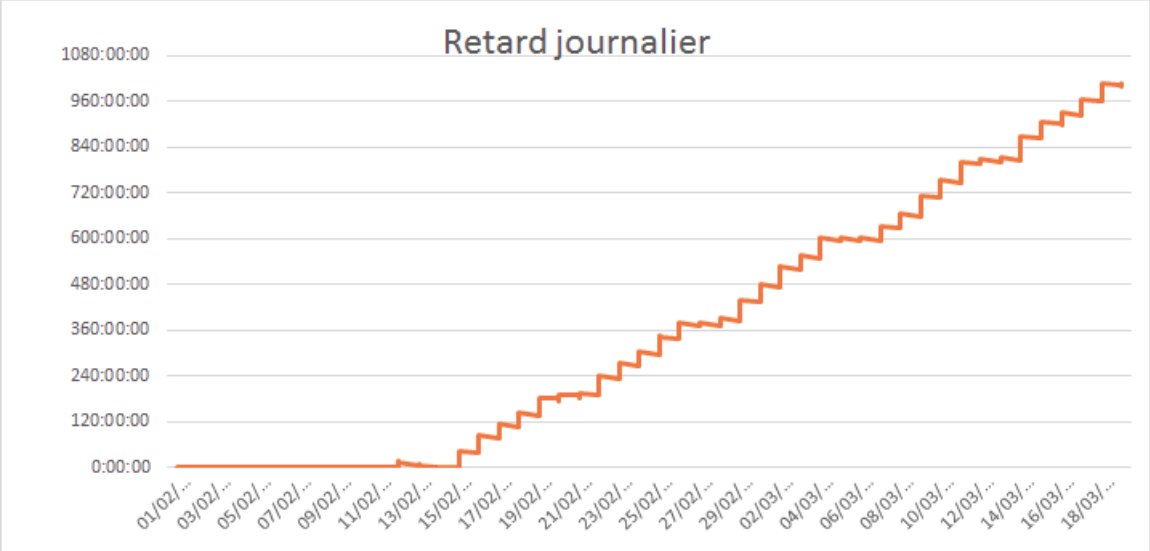

FIGURE 6.6 – Retard avec 8 portes

6.1.4 Aéroport à 12 portes d'embarquement

La répartition des retards pour un aéroport à 12 portes,en pourcentage d'avions dans des intervalles de retards est donnée en figure 6.7.

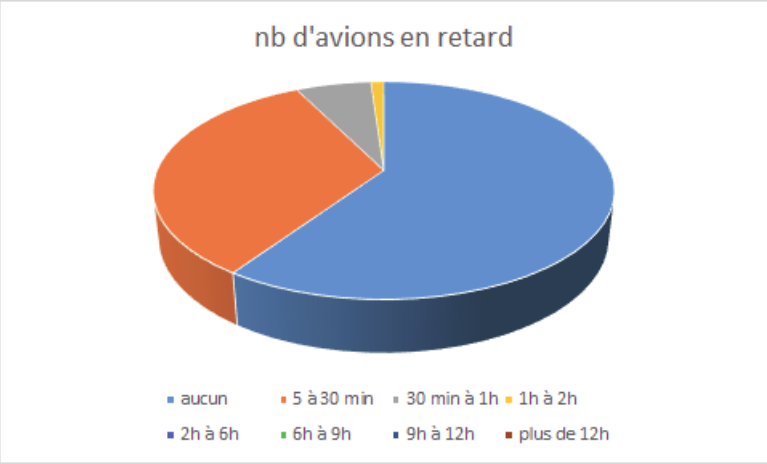

FIGURE 6.7 – Répartition des retards, avec 12 portes

Enfin, la répartition des retards par jour est donnée par la figure 6.8.

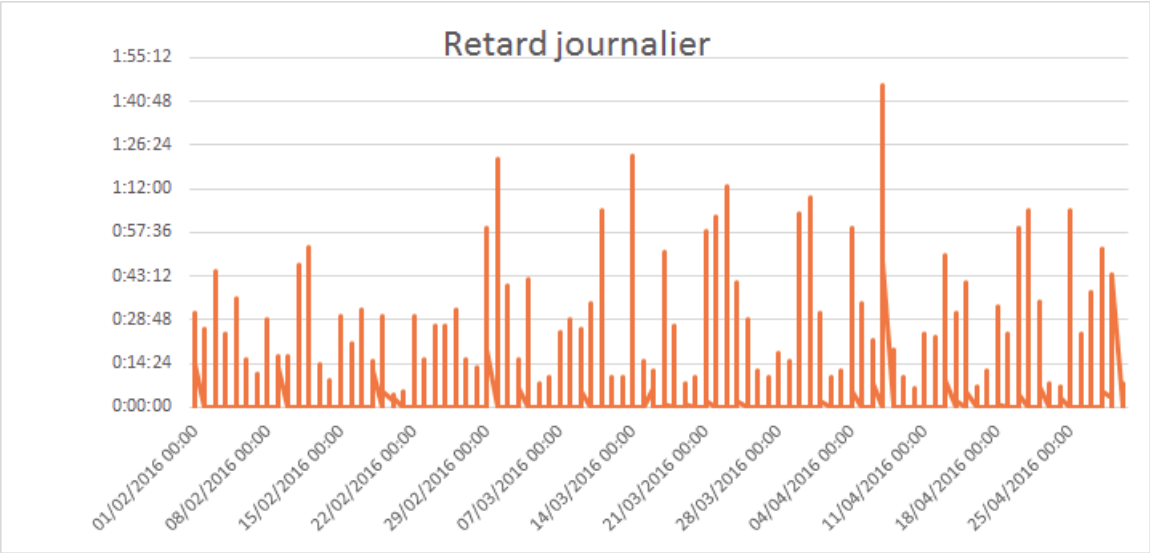

FIGURE 6.8 – Retard avec 12 portes

6.2 Répartition journalière des arrivées

Quelque soit le nombre de portes, les nombres moyens d'atterrissages par heure – respectivement en jour de semaine et en week-end– restent les mêmes, tels qu'on peut les voir dans les figures 6.9 et 6.10.

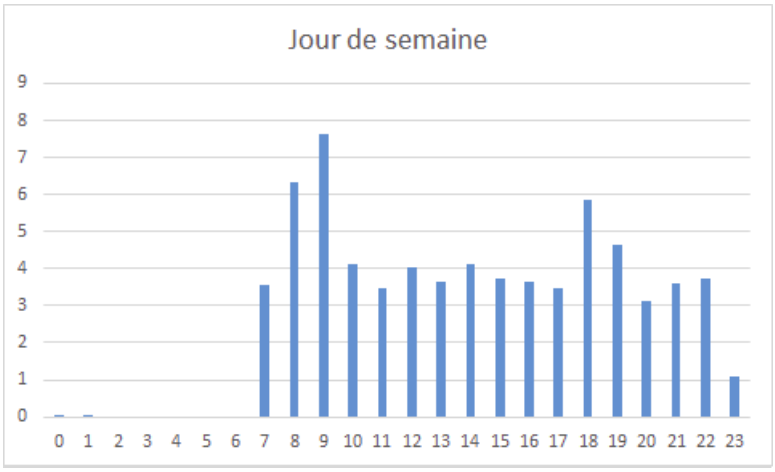

FIGURE 6.9 – Quantité d'atterrissage par horaires en semaine

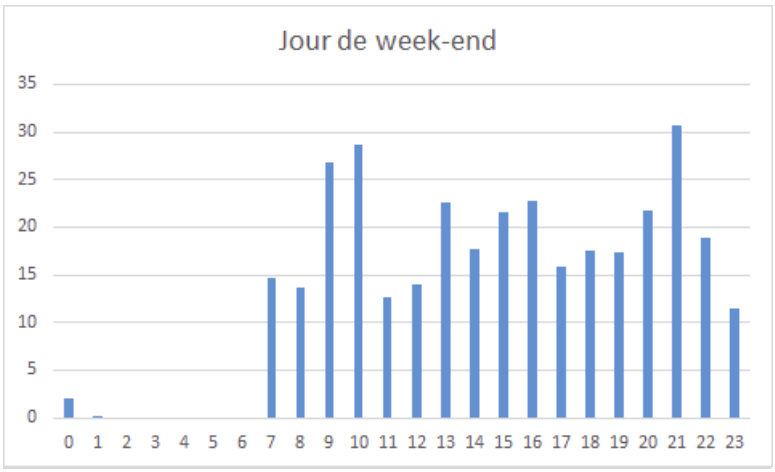

FIGURE 6.10 – Quantité d'atterrissage par horaires le week-end

Chapitre 7

Analyse des résultats

7.1 Un aéroport saturé

Au vu des résultats précédents, il semble évident que l'aéroport tel qu'il est simulé dans les cas à 4, 6 et 8 portes (nous reviendrons sur le cas à 12 portes plus loin) ne fonctionne pas de façon optimal. Les retards s'accumulent très vite, d'autant plus vite que le nombre de portes est faible. Puisque le système de l'aéroport n'est pas capable de gérer les avions plus vite qu'ils n'arrivent, ceux-ci s'accumulent, tout d'abord dans les terminaux puis dans la liste d'attente pour l'atterrissage. Lorsque la nuit tombe et que les avions cessent d'arriver à l'aéroport, la liste commence à se vider, tous les avions en attente se posant sans être remplacés. On tombe alors dans deux cas : soit la liste est suffisamment courte pour que tous les avions se posent avant la réouverture de l'aéroport, auquel cas le lendemain l'aéroport peut reprendre son activité telle quelle, soit la nuit entière ne suffit pas à vider la file, auquel cas les premiers avions arrivant au matin ne peuvent pas atterrir et passent dans la file d'attente qui commence à grandir à chaque jour, entraînant des retards de plus en plus long. Les week-ends, avec leur fréquentation moindre, peuvent aider à désengorger le flux d'avions en attente mais ne sont en général pas suffisant. Une bonne illustration de ce phénomène est la figure 6.6. En effet, jusqu'au 15/02, les retards restent faibles : on observe des pics mais ils sont résorbés avant la fin de la journée. Toutefois, à partir le 15/02, le flux devient ingérable et les retards augmentent de façon linéaire avec le nombre de jour. Le moment où l'aéroport passe en mode saturé advient d'autant plus tôt que le nombre de porte est faible.

Le cas de l'aéroport à 12 portes est particulier et peut apporter un début de réponse. L'aéroport ne passe pas en mode saturé au cours des 90 jours de simulation. Il semble que le rôle de tampon joué par les portes d'embarquement permettent de fluidifier suffisamment le trafic pour qu'il n'y ai jamais de blocage et donc de saturation.

7.2 Des solutions au problème

Nous avons vu à la section précédente qu'un plus grand nombre de porte d'embarquement permet de fluidifier le système. La solution la plus évidente consiste donc à construire plus de portes d'embarquement. Cependant à partir d'un certain nombre de portes c'est le nombre de taxiway qui devient le paramètre bloquant. En effet, si on a assez de place pour répartir les avions au niveau des portes, il faut ensuite avoir suffisamment de taxiway pour que les avions rejoignent ces dites portes rapidement.

Toutefois d'autres solutions peuvent être envisagées. Bien que non testée dans notre simulation, il nous semble important d'en faire part ici, afin que des études puissent être menées dans ces directions. Une des solutions alternatives face à un trafic trop important est de réduire ce trafic. Réduire le nombre d'avions arrivant à l'aéroport permettrai de mieux traiter chaque avion. Nous sommes bien conscient que la tendance actuelle n'est pas au ralentissement du trafic aérien, mais il nous paraissait important de l'évoquer.

Une autre solution serait l'ajout d'une seconde piste d'atterrissage. Une piste supplémentaire permettrait en effet plus de souplesse dans la gestion des flux.

Chapitre 8

Perspectives d'évolutions

8.1 Amélioration de la simulation

Une des pistes d'amélioration de cette simulation serait de mettre en place un logger plus poussé. En effet le logger actuel crée un fichier Excel "brut", qu'il convient de retraiter pour en extraire les données statistiques, ce qui est particulièrement pénible, notamment lorsque l'on n'a aucune formation Excel. Un logger plus poussé contiendrait déjà les formules permettant de calculer les résultats demandés sans manipulation supplémentaire, ou très peu. Une autre possibilité serait d'améliorer la modularité du la simulation, d'autoriser plus de réglages différents, notamment au niveau du nombres des infrastructures (pistes et taxiway) et des fréquences d'arrivées des avions.

8.2 Amélioration du modèle

Quelques améliorations pourraient rendre le modèle plus robuste. Par exemple, il faudrait mettre en place une gestion du détournement des avions : dans le cas où un avion reste en attente au-delà d'un certain temps, il devrait être redirigé vers un autre aéroport et ne plus tourner dans l'espace aérien sur des réserves de carburant infinies. Ces détournements pourraient même advenir de façon préventive, si l'aéroport est saturé, sans que l'avion n'ai a patienter en vol avant de partir vers un autre aéroport.

Chapitre 9

Conclusion

Il semble que l'aéroport souffre d'un problème de temps de traitement des avions : les avions arrivent plus vite qu'ils ne repartent. Un plus grand nombre de portes permettent de gérer ces avions jusqu'à la nuit et de les faire repartir, mais le problème tient au fait que la fréquence d'arrivée en heure de pointe est plus grande que la fréquence possible de départ. La solution d'un plus grand nombre de porte est donc possible, mais il nous semble qu'il est préférable de repenser le système afin de paralléliser les traitements, en ajoutant un deuxième piste-taxiway par exemple. Toutefois, cette proposition ne pourrait être validée qu'avec d'autres tests en simulation.