

POLITECNICO DI TORINO

Dipartimento di Ingegneria Gestionale e della Produzione

Corso di Laurea in Ingegneria Gestionale
Classe L-8

Tesi di Laurea

LATE Lightweight Airport Testing Environment



Relatore
prof. Fulvio Corno

Candidato
Augusto Colongo

AA. 2019/2020

LATE

(Lightweight Airport Testing Environment)

Augusto Colongo - matr.246235

Luglio 2020

Indice

1	Proposta di progetto	3
1.1	Titolo della proposta	3
1.2	Descrizione del problema proposto	3
1.3	Descrizione della rilevanza gestionale del problema	3
1.4	Descrizione dei data-set per la valutazione	3
1.5	Descrizione preliminare degli algoritmi coinvolti	4
1.6	Descrizione preliminare delle funzionalità previste per l'applicazione software	4
2	Introduzione	5
3	Contesto di riferimento	6
3.1	Contesto economico	6
3.2	Contesto operativo	6
3.3	Approccio simulativo	7
4	Dataset	8
4.1	Informazioni aeroportuali	8
4.2	Informazioni sugli aeromobili	8
5	Applicazione e algoritmi	11
5.1	Struttura progetto	11
5.2	Algoritmo simulativo	16
5.3	Algoritmo di pathfinding	21
6	Interfaccia dell'applicazione	22

7	Esempio	25
7.1	Parametri della simulazione	25
7.2	Analisi complessiva	25
7.3	Analisi nel dettaglio	26
8	Conclusioni e possibili sviluppi	29

Elenco delle figure

1	Package LATE	12
2	Package model	13
3	Package airportModel	14
4	Package planeModel	15
5	Package dataImport	16
6	Package DAO	16
7	Arrivo e atterraggio	18
8	Spostamenti e operazioni di terra	19
9	Spostamenti e decollo	20
10	Schermata di avvio	23
11	Schermata di avvio dopo il caricamento del modello aeroportuale	23
12	Schermata dei risultati	24
13	Esempio Torino Caselle: intertempi	26
14	Esempio Torino Caselle: cycle time	27
15	Esempio Torino Caselle: throughput simulazione 9	27
16	Esempio Torino Caselle: throughput simulazione 45	28
17	Esempio Torino Caselle: situazione gates simulazione 45 . . .	28

Elenco delle tabelle

1	Contenuto importato dal formato .apt	9
2	Campi MySQL utilizzati da LATE	10
3	Schema dei package	11

1 Proposta di progetto

1.1 Titolo della proposta

LATE (Lightweight Airport Testing Environment)

1.2 Descrizione del problema proposto

L'applicazione intende fornire uno strumento per il calcolo del traffico massimo che un aeroporto può gestire, attraverso un approccio simulativo in cui verranno effettuati degli stress test. L'obiettivo della simulazione è quello di fornire all'utente dei KPI (*Key Performance Indicator*) atti ad analizzare la risposta dell'infrastruttura aeroportuale a valori diversi di input di traffico.

1.3 Descrizione della rilevanza gestionale del problema

Il calcolo della capacità di un'infrastruttura è di notevole rilevanza progettuale e gestionale. Nel primo caso i benefici sono lampanti: in fase di progettazione si devono garantire dei minimi livelli di servizio in termini di efficienza dell'infrastruttura. In fase di gestione, questo strumento potrebbe aiutare la valutazione dell'impatto di manutenzioni e imprevisti. Un esempio pratico potrebbe essere la chiusura di una parte di un terminal per ristrutturazione: in tal caso si potrebbero escludere dal modello i gate coinvolti, lanciare la simulazione e valutare l'impatto dell'evento sull'efficienza aeroportuale.

1.4 Descrizione dei data-set per la valutazione

I dati che utilizzerà l'applicazione sono:

- Dati sull'infrastruttura aeroportuale:

i dati verranno importati dai modelli aeroportuali creati dalla comunità del simulatore X-Plane che si possono trovare e scaricare gratuitamente dal sito <https://gateway.x-plane.com/>. Tutti gli scenari presenti sul sito sono gratis e pubblicati sotto la licenza GPLv2. L'importazione di questi scenari fornirà la mappa delle piste, delle taxiways, dei gate e le principali procedure per l'arrivo e la partenza degli aeromobili. Inoltre, X-Plane ha rilasciato un software per la creazione di questi scenari che permetterà all'utente di crearne di immaginari o di apportare delle modifiche a quelli esistenti;

- Dati sugli aerei:

i dati sugli aerei verranno presi da un catalogo dell'FAA (*Federal Aviation Administration*) che si può trovare al seguente link: https://www.faa.gov/airports/engineering/aircraft_char_database/. I dati che ci interessano sono principalmente le dimensioni degli aeromobili e le loro velocità di atterraggio.

Il traffico non sarà basato su quello reale, in quanto lo scopo dell'applicazione non è di simulare una situazione preesistente, ma di sottoporre l'infrastruttura a carichi variabili in modo da calcolarne la capacità massima.

1.5 Descrizione preliminare degli algoritmi coinvolti

I principali algoritmi coinvolti saranno:

- un algoritmo simulativo a eventi discreti che costituirà la parte principale dell'applicazione: simulerà l'interazione degli aeroplani con il modello aeroportuale;
- un algoritmo di pathfinding per fornire agli aerei un percorso da seguire per arrivare al gate e per ritornare alla pista, una volta effettuate le operazioni di sbarco e di imbarco;

1.6 Descrizione preliminare delle funzionalità previste per l'applicazione software

L'applicazione sarà strutturata nel seguente modo:

- nella prima pagina l'utente sarà in grado di:
 - Definire un percorso file per l'importazione del modello aeroportuale;
 - Impostare la proporzione in cui verranno scelti i vari modelli di aerei per la simulazione;
 - Impostare altre variabili della simulazione;
 - Dopo l'importazione del file, all'utente verranno presentati dei dati sull'aeroporto;
- nella seconda pagina più schermate proporranno i vari KPI e grafici che aiuteranno l'utente all'analisi dello stress test.

2 Introduzione

Sin dall'infanzia, e anche grazie alle numerose possibilità avute di viaggiare, sono sempre stato affascinato dalle infrastrutture, che fossero aeroporti, ferrovie o autostrade.

Le mie insegnanti della scuola primaria mi trovavano spesso intento a tracciare punti, simboli e linee, apparentemente a caso, su un foglio. Ciò che sembrava casuale, aveva in realtà un senso: le prime erano città, i secondi ostacoli naturali e le terze le infrastrutture di varia natura che le connettevano.

Se queste sono sommariamente le origini del mio interesse per le infrastrutture, con l'avanzare della tecnologia e degli anni, ho coltivato questa passione unendovi un aspetto ludico: alle linee, ai simboli e ai punti si sono sostituiti i più tecnologici simulatori (Microsoft Flight Simulator, Train Simulator, Euro Truck Simulator) e video game gestionali/manageriali come Open Transport Tycoon.

Le numerose ore di gioco mi hanno fatto approcciare, da una prospettiva di "utente virtuale", a quelle che sono oggi le basi per la simulazione che presenterò in questa tesi.

All'università, grazie agli esami di informatica, ho potuto dedicarmi alla programmazione e, come primo progetto personale, ho creato un programma che gestisse il traffico di un aeroporto. Essendo tuttavia il linguaggio C, nel quale programmavo, di basso livello, il risultato¹ è stato limitato nelle funzionalità.

Al momento di scegliere la tesi, grazie alle competenze acquisite studiando per gli esami *"Tecniche di Programmazione"* e *"Programmazione e Gestione della Produzione"*, ho potuto dare concretezza alla mia idea. Da questo nasce il progetto LATE, che aggiunge all'idea originale un livello di analisi, senza il quale non avrebbe potuto essere inquadrato nell'ambito gestionale.

¹ Github: <https://github.com/augustocolo/AirportManager>.

3 Contesto di riferimento

3.1 Contesto economico

Quello dell'*Air Traffic Management* (ATM) è un mercato in prevista forte crescita, prevalentemente grazie agli investimenti nel comparto tecnologico: si stima infatti un CAGR (*Compound Annual Growth Rate*) del 4.20%, da 14.1 miliardi di dollari nel 2018 a 18.8 miliardi nel 2025, soprattutto nell'area dell'Asia-Pacifico[1].

L'ATM è diviso in due rami: *airspace* e *ground*. Il primo riguarda tutte le operazioni relative al traffico “in aria”, quali l'*Air Traffic Flow Management* e l'*Airspace Management*, il secondo si occupa della gestione delle operazioni nelle fasi a terra (*Airport Operations Management*).

I nuovi strumenti tecnologici dell'ATM vengono impiegati a supporto sia di investimenti *greenfield* (in un'area precedentemente non utilizzata), che *brownfield* (in un'area oggetto di ammodernamento/riconversione).

Per quanto concerne l'impiego di tali tecnologie in investimenti *greenfield*, caso emblematico è quello della costruzione del nuovo aeroporto di Istanbul Havalimani (operativo da aprile 2019) per cui, grazie a moderni strumenti tecnologici, sono state ridisegnate, ovviamente ottimizzandole, le rotte sino a Malmö, in Svezia.

Un'applicazione in ambito *brownfield* è quella che riguarda Milano Linate, i cui i lavori di ammodernamento (creazione di nuovi spazi per gli utenti, di nuovi gate e rifacimento di una pista) in una prima fase (luglio-ottobre 2019) hanno dirottato il traffico su Milano Malpensa, per poi consentire la riapertura di nove gate (ottobre 2019) fino ad un completamento delle diverse infrastrutture e funzionalità (marzo 2021).

3.2 Contesto operativo

L'ecosistema dell'*Airport Operations Management*, molto vasto, riguarda la gestione di:

- traffico di terra
- passeggeri
- risorse del ground handling
- arrivi e partenze
- manutenzione

Dotarsi dei più evoluti sistemi tecnologici in tutti questi comparti è essenziale per gli *stakeholders* dell'aviazione:

- per le compagnie aeree, perché permette di ridurre al minimo il *turnaround time* (o tempo di rotazione): il periodo che trascorre dall'atterraggio di un aereo al suo successivo decollo, nel quale non genera reddito ma solo costi (tasse aeroportuali, costi di ammortamento, costi fissi). Una caratteristica delle compagnie low cost è, infatti, la massima riduzione del turnaround time;
- per i gestori degli aeroporti, perché permette di aumentare l'efficienza dell'infrastruttura, incrementando il numero di aerei che possono essere processati nello stesso tempo;
- per i contractor aeroportuali cioè coloro che forniscono servizi all'aeroporto, perché possono fare stime precise sulle proprie necessità, in termini di personale e di mezzi.

È quindi necessario dotarsi di sistemi che possano prevedere la capacità di un aeroporto, in modo da poter valutare l'efficienza delle infrastrutture e adottare accorgimenti giusti per migliorarla.

3.3 Approccio simulativo

I sistemi attualmente in commercio sono di tipo simulativo: un ottimo prodotto è CAST² di Airport Research Center, sviluppato in collaborazione con Eurocontrol e alcuni tra i maggiori aeroporti europei.

Una loro caratteristica comune è l'utilizzo di scenari preimpostati derivati da situazioni di traffico reali. Questo approccio sconta tuttavia il limite di non fornire informazioni precise sulla capacità teorica massima dell'aeroporto preso in considerazione. Ciò che invece LATE si propone di fare, è applicare la tecnica dello stress testing, sottoponendo il modello aeroportuale ad un aumento progressivo dei carichi per constatarne la risposta. Meno pratico e più teorico, non avendo fini commerciali, il mio approccio è più vasto, esamina più possibilità e dà una visione più ampia: stessa metodologia simulativa, obiettivi simili, approccio differente.

² CAST: <https://arc.de/cast-express-simulation/>.

4 Dataset

4.1 Informazioni aeroportuali

Come già detto in precedenza, LATE ottiene le informazioni sull'aeroporto che deve simulare a partire da un formato open-source sviluppato dalla Laminar Research, software-house che si occupa dello sviluppo del simulatore X-Plane.

La comunità del videogioco è molto attiva, perciò quasi tutti gli aeroporti sono creati dagli utenti stessi. Dopo una fase di testing, i nuovi aeroporti sono pubblicati sul sito X-Plane Gateway e vengono inclusi nelle release successive del software.

I creatori degli scenari utilizzano il software World Editor che, mediante un'interfaccia grafica, genera un file in formato .apt le cui specifiche sono riportate online³. La versione utilizzata è la 1130, pubblicata l'11 Aprile 2019. Il file va interpretato riga per riga, ognuna inizia con un **row code** che specifica il tipo di informazione che lo seguirà. (*Vedere tabella 1*)

LATE importa solo alcune informazioni, utilizzando esclusivamente quelle funzionali allo svolgimento della simulazione e tralasciando, per esempio, quelle di natura grafica. Inoltre, durante l'importazione calcola delle informazioni derivate, tra le quali la lunghezza delle piste e delle taxiway.

4.2 Informazioni sugli aeromobili

Le informazioni sugli aeromobili utilizzate da LATE provengono da un database⁴ fornito dalla FAA (*Federal Aviation Administration*), agenzia che si occupa della regolamentazione dell'aviazione civile statunitense. Il database è incluso nell'appendice 1 del regolamento per la progettazione di aeroporti civili e contiene le caratteristiche dei modelli autorizzati a volare negli Stati Uniti.

Come prima cosa ho provveduto a pulire la tabella: ho eliminato tutte le righe con campi mancanti, riducendo il numero di aerei da 873 a 317. Per semplificare l'utilizzo dell'applicazione ne ho selezionati 26, che ho evidenziato con il campo **Relevant**, e che potranno essere scelti dall'utente.

Essendo un database statunitense, il compito successivo è stato quello di convertire le unità di misura dal sistema imperiale/aeronautico al sistema

³ Specifiche formato apt: http://developer.x-plane.com/wp-content/uploads/2019/04/XP-APT-1100-Spec_revised_04-11-2019.pdf.

⁴ Database FAA: https://www.faa.gov/airports/engineering/aircraft_char_database.

Tabella 1: Contenuto importato dal formato .apt

Row code	Gruppo	Contenuto	Informazioni
1	Aeroporto	Informazioni	Elevazione (in piedi) Codice ICAO Nome
1302	Aeroporto	Metadati	Codice IATA Città Nazione
100	Pista di atterraggio	Informazioni	Numero Coordinate geografiche Larghezza
1000	Traffic flow	Identificativo	
1110	Traffic flow	Pista utilizzata	Numero Utilizzo (Arrivi/Partenze)
1201	Taxiways	Nodo	Identificativo Coordinate geografiche
1202	Taxiways	Arco	Identificativi nodi agli estremi Direzione
1300	Gate	Informazioni	Identificativo Coordinate geografiche Tipologia (Gate/Tie down)
1300	Gate	Metadati	Dimensioni Tipologia di traffico

metrico (e questo per tutti i campi, anche quelli che non sono impiegati da LATE). Ho utilizzato i seguenti fattori:

$$\begin{aligned} 1 \text{ m/s} &= 1.94384 \text{ kts} \\ 1 \text{ m} &= 3.28084 \text{ ft} \\ 1 \text{ kg} &= 2.2046 \text{ lb} \\ 1 \text{ m}^2 &= 10.763 \text{ sqft} \end{aligned}$$

Infine ho importato, utilizzando il software DataGrip dell'azienda JetBrains, il file, fornito in formato .xlsx, in un database MySQL, il cui script di inizializzazione è disponibile al link: https://github.com/TdP-prove-finali/ColongoAugusto/blob/master/Database/faa_database_Aircraft.sql.

Nella tabella 2 sono riportati i campi dello schema *Aircraft* presente sul database MySQL.

Tabella 2: Campi MySQL utilizzati da LATE

Campo MySQL	Tipo MySQL	Descrizione
Manufacturer	VARCHAR(64)	Nome del produttore
Model	VARCHAR(64)	Nome del modello
Relevant	TINYINT(1)	Flag per filtrare gli aerei più significativi
ApproachSpeedMetersPerSecond	DOUBLE	Velocità di avvicinamento alla pista
WingspanMeters	DOUBLE	Apertura alare
MaximumTakeOffWeightKilograms	DOUBLE	Peso massimo di decollo

5 Applicazione e algoritmi

5.1 Struttura progetto

Il progetto LATE contiene 7 package, composti da 22 classi e 4 enum.

Ulteriori informazioni sul contenuto dei package si trovano nella tabella 3, mentre i diagrammi UML delle classi sono riportati nelle figure 1, 2, 3, 4, 5 e 6.

Tabella 3: Schema dei package

Nome package	# classi	# enum	Contenuto
LATE	4	0	Interfaccia grafica
model	3	1	Simulazione
airportModel	6	3	Modello aeroportuale
dataImport	2	0	Importazione modello aeroportuale
planeModel	4	0	Aerei e voli
DAO	3	0	Connessione al database degli aerei
utility	1	0	Utility functions

Figura 1: Package LATE

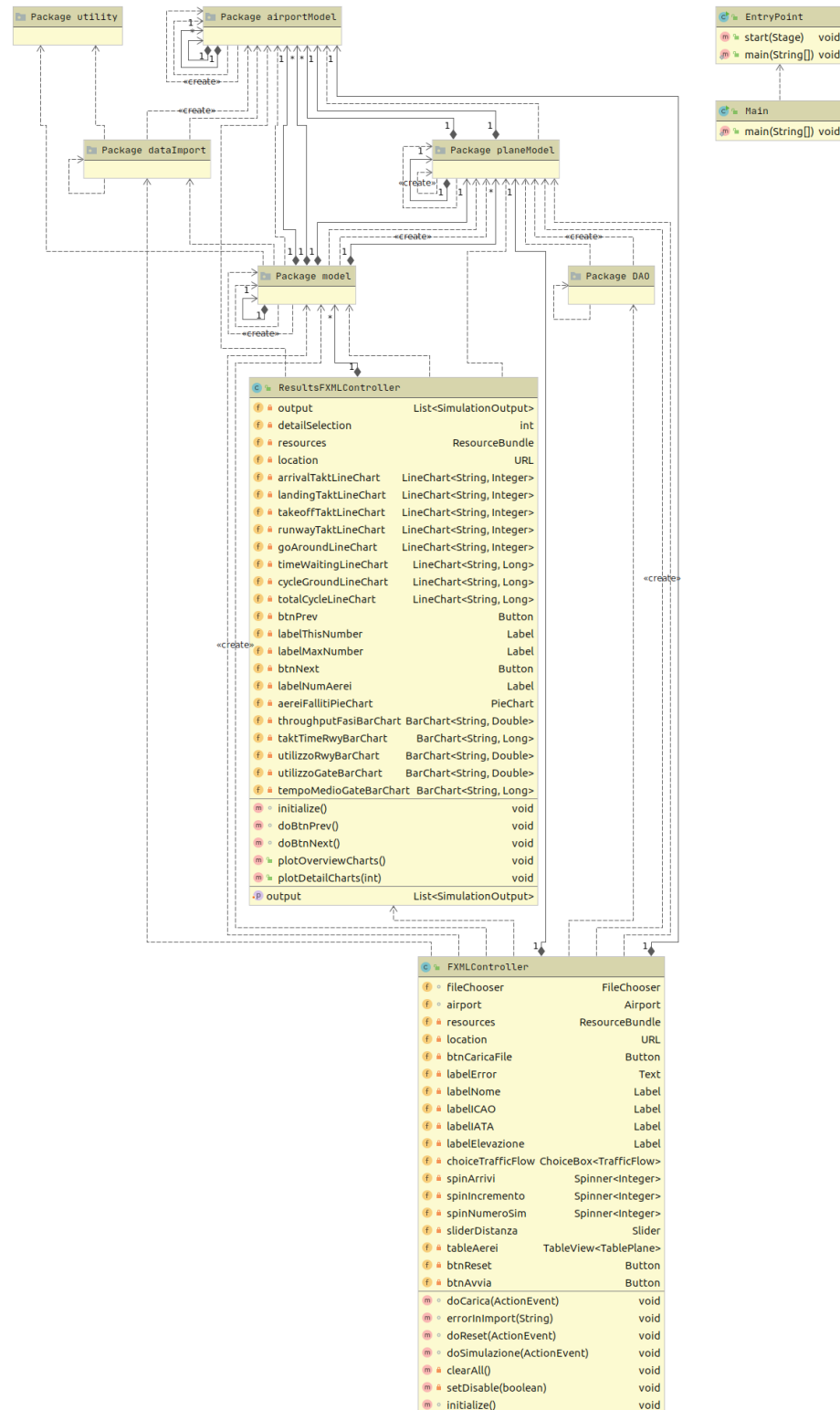


Figura 2: Package model

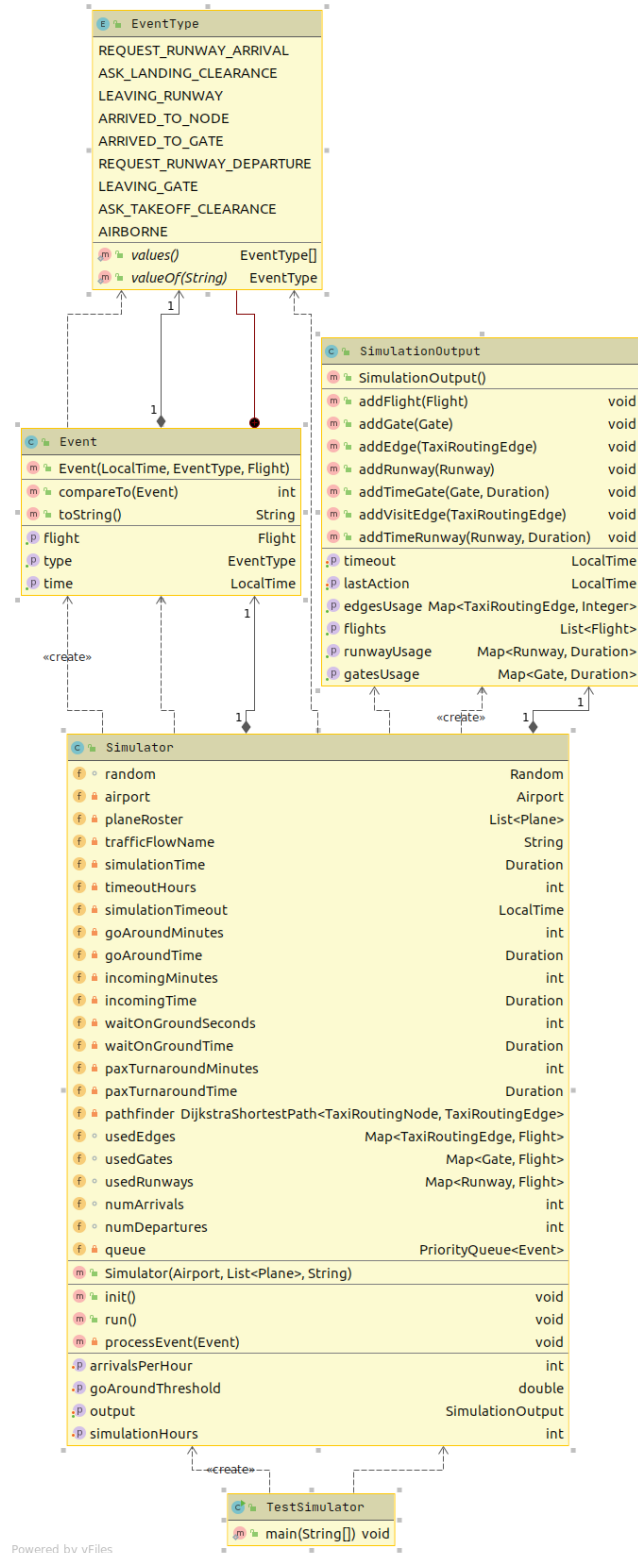


Figura 3: Package airportModel

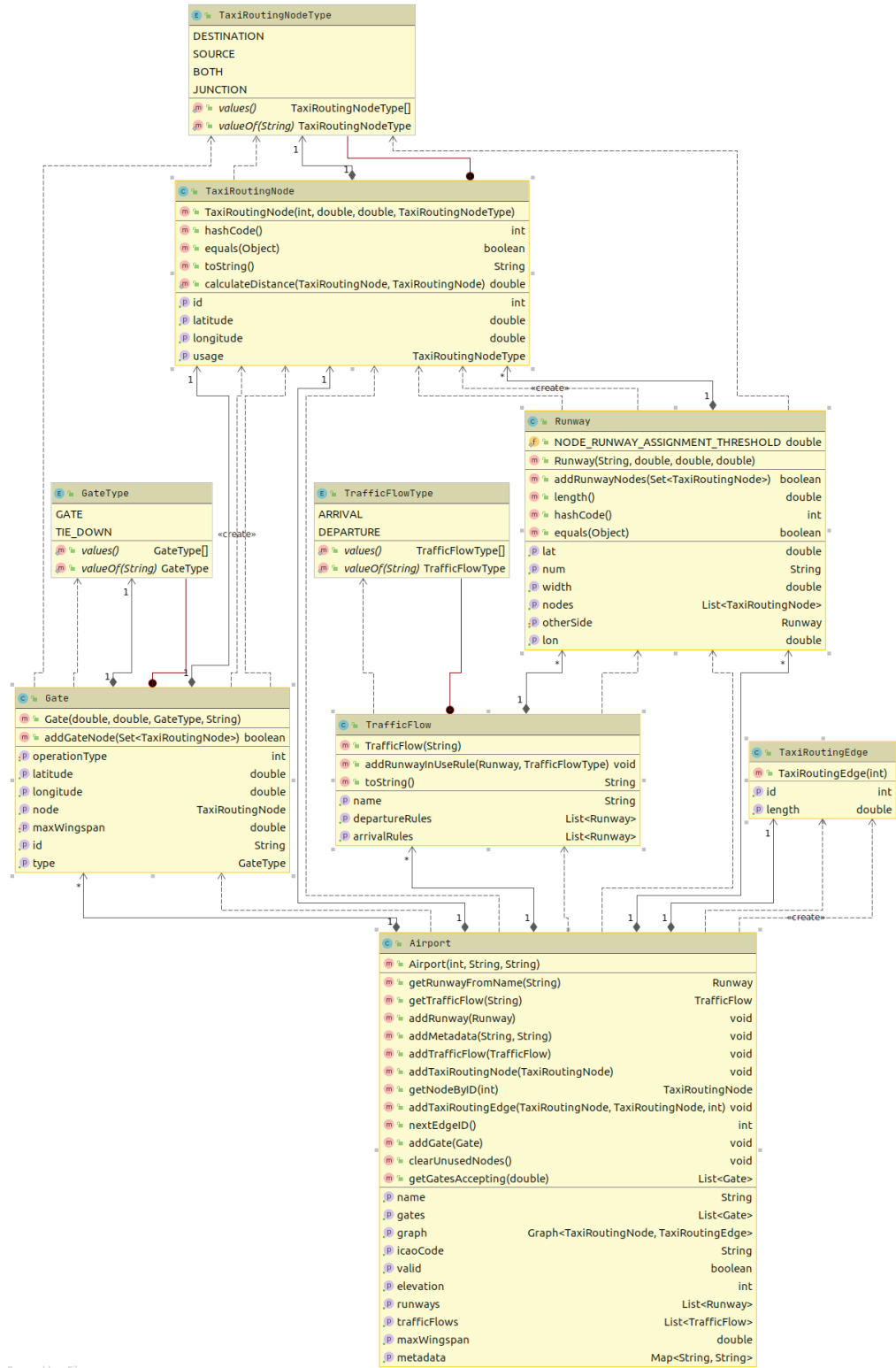


Figura 4: Package planeModel

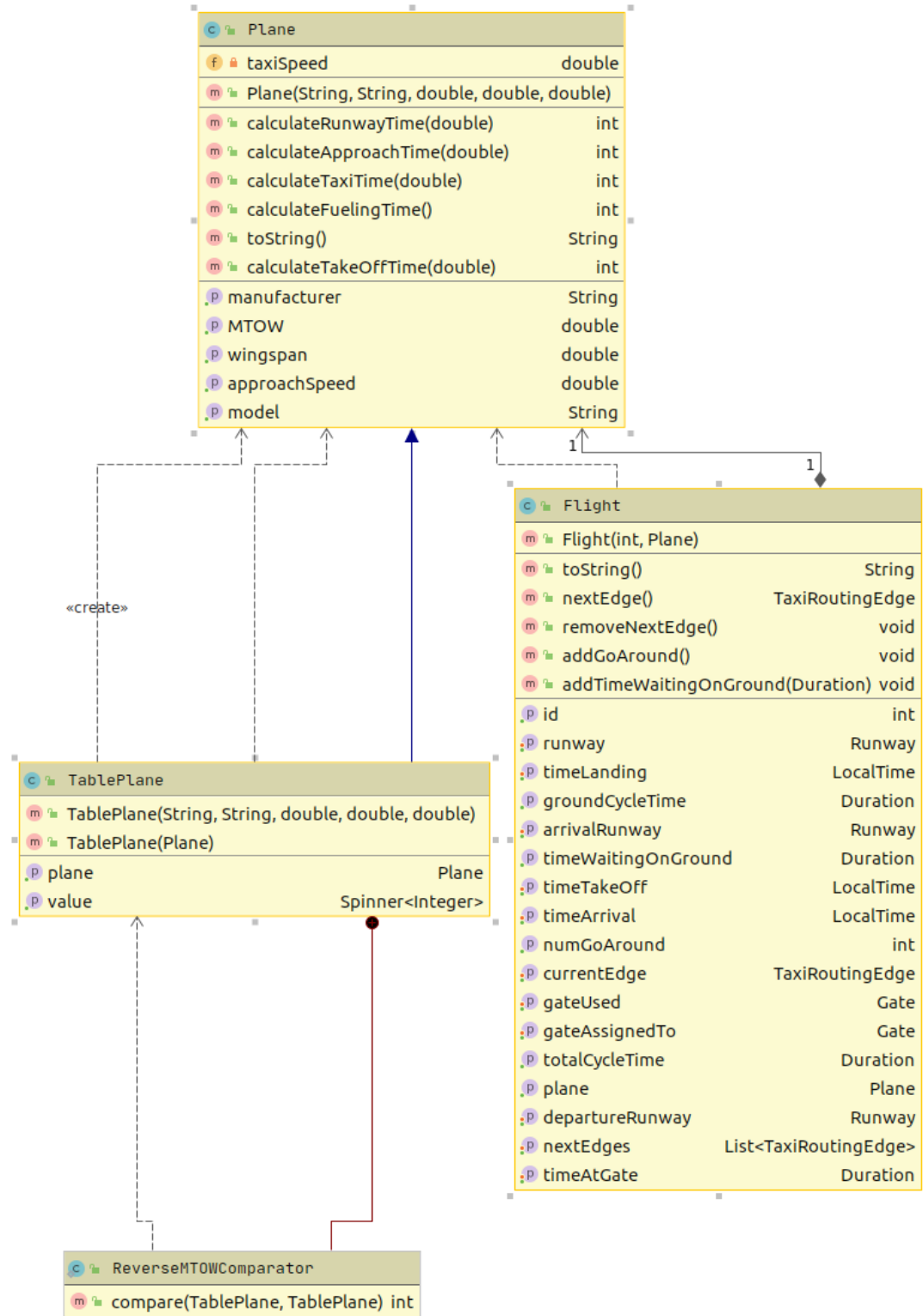


Figura 5: Package dataImport

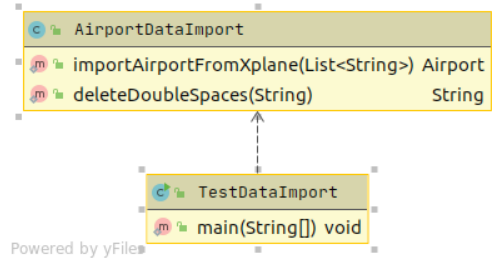
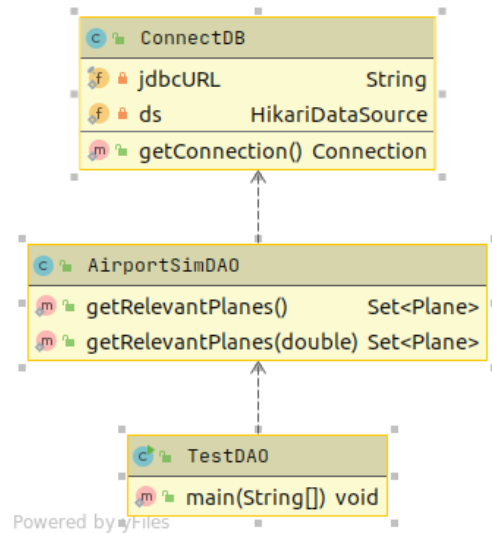


Figura 6: Package DAO



5.2 Algoritmo simulativo

Ho scelto per questa simulazione un algoritmo a eventi discreti in cui gli aerei (rappresentati dalla classe *Flight*) entrano ed escono dal sistema per mezzo degli eventi *REQUEST RUNWAY ARRIVAL* e *AIRBORNE*.

I parametri della simulazione non modificabili attraverso l'interfaccia grafica sono:

- Simulation time: 23 ore (finestra di simulazione, ovvero dopo quante ore simulate il programma andrà in timeout);
- Go around time: 10 minuti (durata dell'azione di Go Around);
- Incoming time: 10 minuti (separazione temporale tra gli eventi *REQUEST RUNWAY ARRIVAL* e *ASK LANDING CLEARANCE*);

- Wait on ground time: 20 secondi (tempo di attesa a terra nel caso in cui non sia possibile eseguire l'azione richiesta);
- Passenger turnaround time ($t_{paxTurnaround}$): 30 minuti (durata delle operazioni di sbarco e imbarco dei passeggeri);
- Taxi speed (v_{taxi}): $20 \text{ kts} = 10.289 \text{ m/s}$ (velocità di rullaggio degli aeromobili);
- Decelerazione media sulla pista (dec_{pista}): 2.012 m/s^2
ottenuta da una media dei dati calcolati nell'articolo "Computer Simulation Model for Airplane Landing-Performance Prediction" [2].

Dai parametri sopra citati e dal modello otteniamo altre misure:

- Landing time ($t_{landing}$): tempo che intercorre tra gli eventi *ASK LANDING CLEARANCE* e *LEAVING RUNWAY*

$$t_{landing} = t_{runway} + t_{approach} \quad (1)$$

$$t_{approach} = \frac{d_{goAround}}{v_{approach}} \quad (2)$$

$$t_{runway} = t_{decelerazione} + t_{taxi} \quad (3)$$

$$t_{decelerazione} = \frac{v_{approach} - v_{taxi}}{dec_{pista}} \quad (4)$$

$$t_{taxi} = v_{approach} \cdot t_{decelerazione} - \frac{1}{2} \cdot dec_{pista} \cdot t_{decelerazione}^2 \quad (5)$$

- Turnaround time ($t_{turnaround}$): tempo delle operazioni dell'aeromobile al gate

$$t_{turnaround} = t_{paxTurnaround} + t_{rifornimento} \quad (6)$$

$$t_{rifornimento} = \frac{m_{carburante}}{D_{carburante}} \cdot v_{rifornimento} \quad (7)$$

$$m_{carburante} \approx 0.17 MTOW_{aereo} \quad (8)$$

in cui $D_{carburante} \approx 0.719 \text{ kg/L}$ [3] e $v_{rifornimento} \approx 50 \text{ L/s}$ [4].

Nelle figure 7, 8 e 9 sono riportati i flowchart del processo di attraversamento dell'infrastruttura.

Figura 7: Arrivo e atterraggio

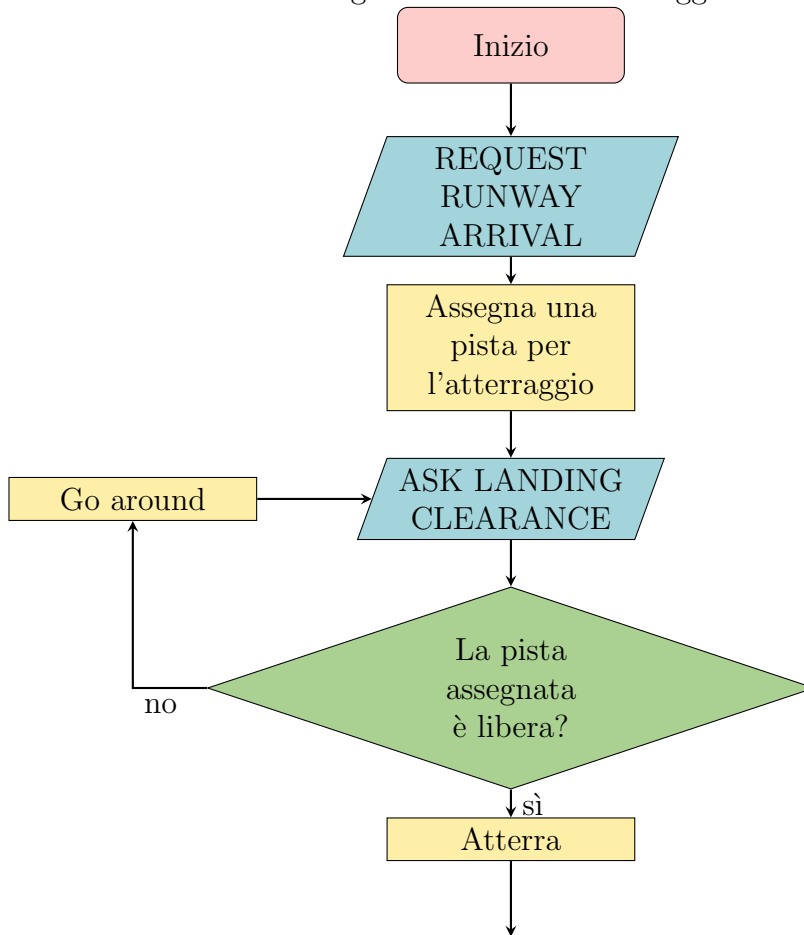


Figura 8: Spostamenti e operazioni di terra

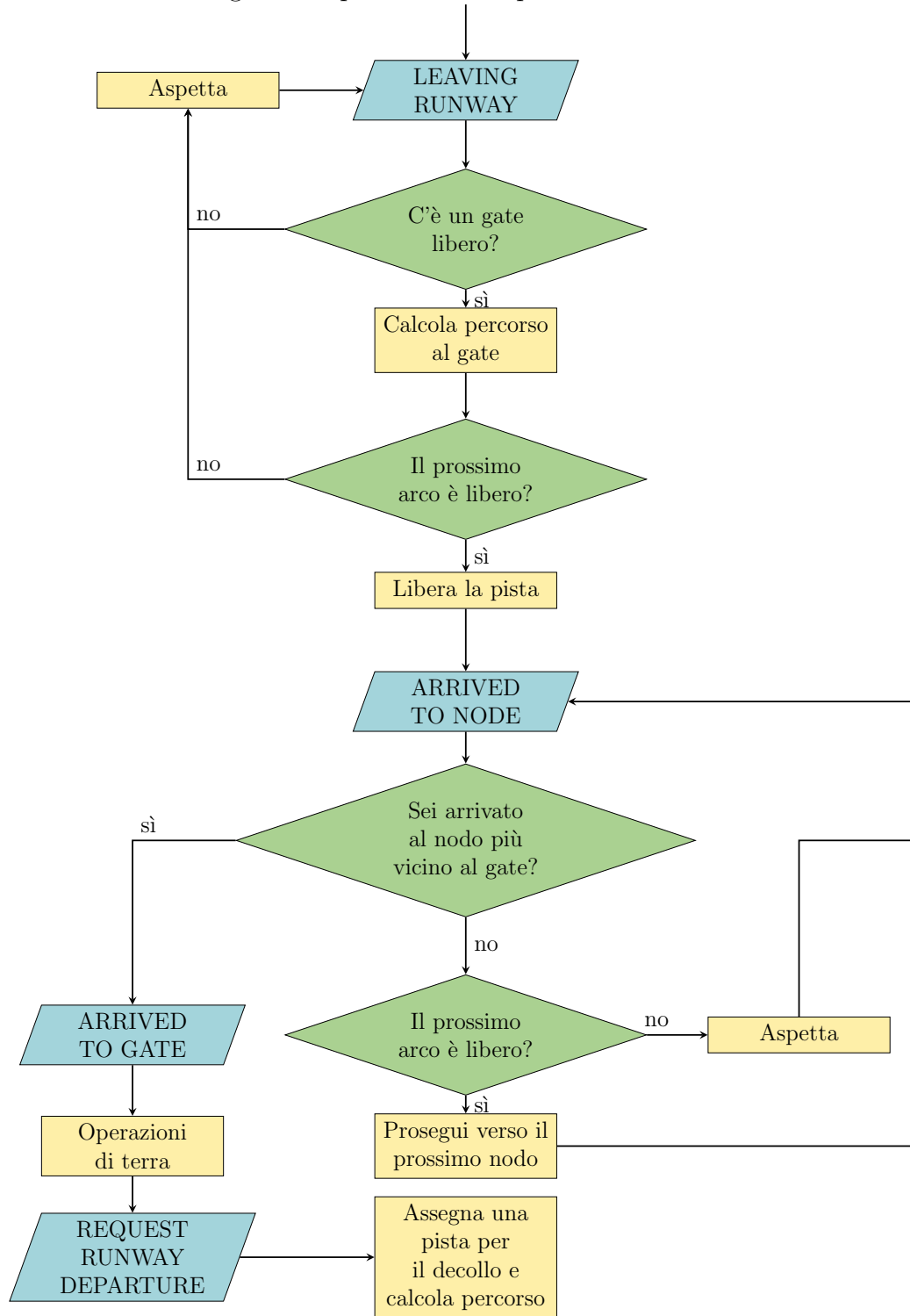
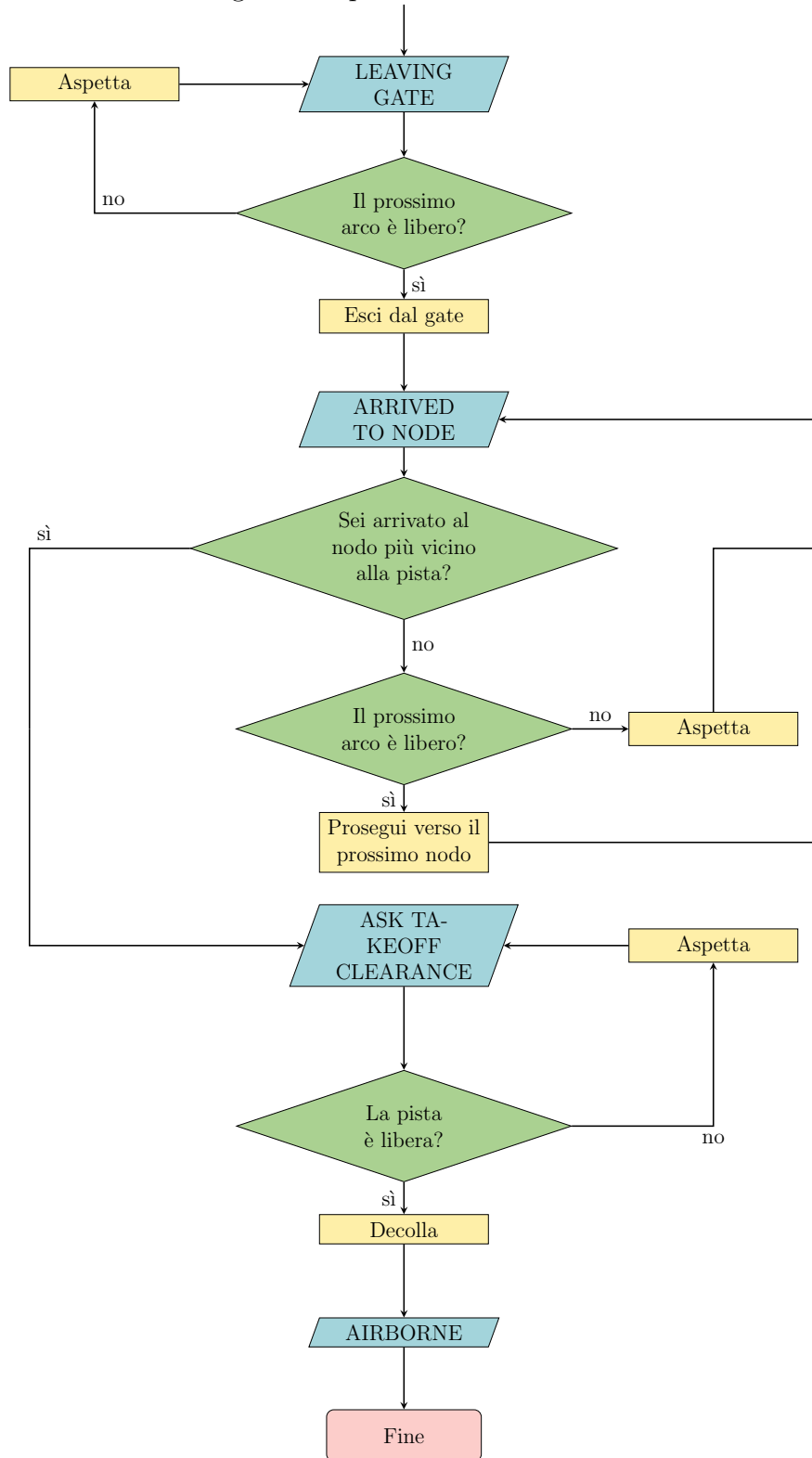


Figura 9: Spostamenti e decollo



5.3 Algoritmo di pathfinding

Gli aerei, per spostarsi dalla pista al gate e viceversa, devono seguire un percorso attraverso il grafo pesato e orientato formato dalle taxiway. Per calcolarlo, ho deciso di utilizzare l'algoritmo *Dijkstra's Shortest Path First*[5], nell'implementazione della libreria jGraphT, il cui metodo *getPath* permette di ottenere un oggetto di tipo *GraphPath* contenente la lista di archi da percorrere.

6 Interfaccia dell'applicazione

L'interfaccia dell'applicazione è composta da due schermate: una in cui si può scegliere il modello aeroportuale (fig.10) e impostare le variabili di simulazione (fig.11), un'altra in cui vengono proposti i risultati (fig.12).

I risultati che vengono presentati all'utente sono di due tipologie:

- la prima formulata per comparare i risultati delle varie simulazioni, divisi in due categorie di competenza:
 - Intertempi: *Tempo di interarrivo nello spazio aereo, Intertempo tra atterraggi medio, Intertempo tra decolli medio, Takt time piste medio;*
 - Cycle time: *Numero di Go Around, Tempo atteso a terra medio, Cycle Time aeroportuale medio, Cycle Time complessivo medio;*
- la seconda per vedere in dettaglio le singole simulazioni, suddivisa in tre categorie di competenza:
 - Generale: *Numero di aerei, Percentuale di voli riusciti, Throughput delle fasi;*
 - Piste: *Takt Time e Utilizzo percentuale;*
 - Gate: *Utilizzo percentuale e Tempo medio di permanenza.*

Figura 10: Schermata di avvio

LATE *Lightweight Airport Testing Environment*

Dati Aeroporto

Nome:
 Codice ICAO:
 Codice IATA:
 Elevazione (s.l.m.):
 Seleziona Traffic Flow:

Parametri di Simulazione

Arrivi per ora iniziali:
 Incremento per ogni simulazione:
 Numero di simulazioni:
 Distanza richiesta di atterraggio:

Aerei

Costruttore	Modello	Apertura Alare	Velocità di at...	MTOW	Valore
No content in table					

Reset Avvia simulazione

Figura 11: Schermata di avvio dopo il caricamento del modello aeroportuale

LATE *Lightweight Airport Testing Environment*

Dati Aeroporto

Nome: Torino Caselle
 Codice ICAO: LIMF
 Codice IATA: TRN
 Elevazione (s.l.m.): 300
 Seleziona Traffic Flow: South flow

Parametri di Simulazione

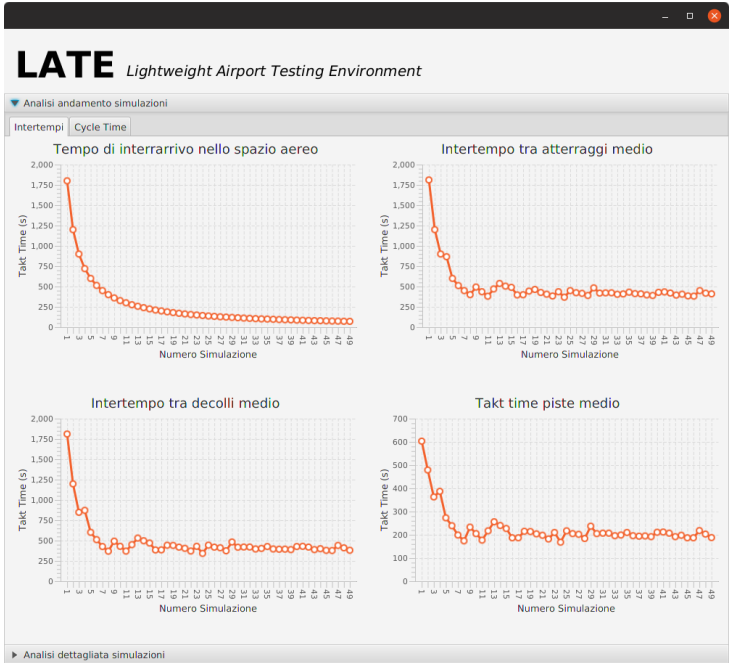
Arrivi per ora iniziali:
 Incremento per ogni simulazione:
 Numero di simulazioni:
 Distanza richiesta di atterraggio:

Aerei

Costruttore	Modello	Apertura Alare	Velocità di at...	MTOW	Valore
Airbus	A321neo	35.8	72.02	93500.86	0
Boeing	737-800	34.32	73.05	79016.6	1
Airbus	A320neo	35.8	69.96	79000.73	1
Airbus	A319neo	35.8	64.82	75500.77	0
Boeing (McDon...	MD-90-30	32.87	70.99	70761.14	0
Embraer	EMB 190 Stand...	28.73	63.53	47790.53	0

Reset Avvia simulazione

Figura 12: Schermata dei risultati



7 Esempio

7.1 Parametri della simulazione

- Modello aeroportuale utilizzato: Torino Caselle⁵
- Traffic flow: South Flow
- Arrivi per ora iniziali: 1
- Incremento per ogni simulazione: 1
- Numero di simulazioni: 50
- Distanza richiesta di atterraggio: 2 *nm*
- Probabilità aerei:
 - Airbus A321neo: 8.33%
 - Boeing 737-800: 25.00%
 - Airbus A320neo: 25.00%
 - Airbus A319neo: 8.33%
 - McDonnell Douglas MD-90-30: 8.33%
 - Embraer 190 Standard: 8.33%
 - British Aerospace 146-300: 8.33%
 - Embraer 170 Long Range: 8.33%

7.2 Analisi complessiva

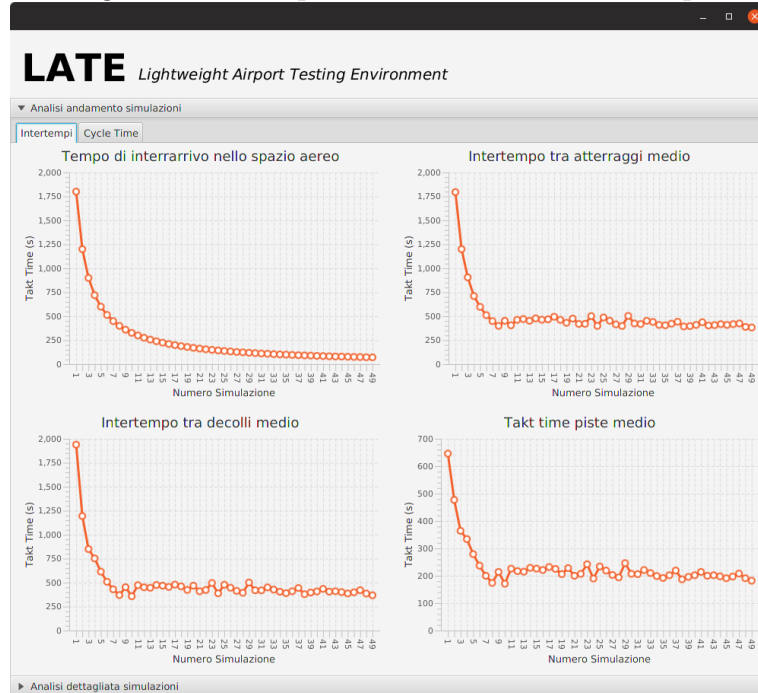
L'analisi dei risultati forniti dall'applicazione deve essere effettuata iniziando dai dati d'insieme contenuti nella scheda *Analisi andamento simulazioni*. Nel caso in questione, possiamo notare che le curve degli intertempi (figura 13) hanno un andamento inversamente proporzionale al numero della simulazione fino alla n°8. A seguire, i tre grafici *Intertempo tra atterraggi medio*, *Intertempo tra decolli medio* e *Takt Time piste medio* fluttuano attorno ad un valore costante.

In figura 14, possiamo vedere come l'andamento dei grafici si modifichi a partire dalla simulazione n°8, ad eccezione del *Cycle Time aeroportuale medio*.

⁵ LIMF by carlosr: <https://gateway.x-plane.com/scenery/page/LIMF>.

L'analisi di queste due schermate ci porta ad una conclusione: la capacità dell'aeroporto Torino Caselle, in queste condizioni, è di circa 8 *aerei/ora*. Inoltre, l'infrastruttura di terra sembra non essere satura, infatti il *Cycle Time aeroportuale medio* non aumenta proporzionalmente al traffico gestito.

Figura 13: Esempio Torino Caselle: intertempi



7.3 Analisi nel dettaglio

Nella scheda *Analisi dettagliata simulazioni* si devono andare a verificare le *assumption* effettuate nel punto precedente.

- **Simulazione numero 9:** il grafico in figura 15 corrobora le ipotesi precedenti, ovvero che il throughput massimo gestibile dall'aeroporto sia di 8 *aerei/ora* e che il bottleneck si trovi tra le fasi *Arrival* e *Landing*.
- **Simulazione numero 45:** il grafico in figura 16 è un'ulteriore conferma, con dati ancora più evidenti, di quanto detto al punto precedente. In figura 17 sono presenti i grafici che ci permettono di analizzare le performance dei gate, non sfruttati appieno (utilizzo compreso tra 10% e 40%).

Figura 14: Esempio Torino Caselle: cycle time

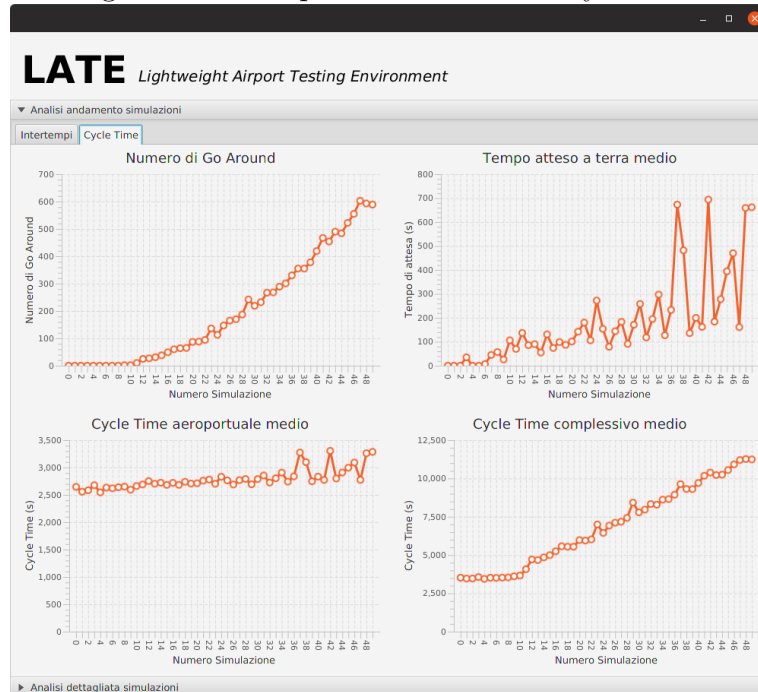


Figura 15: Esempio Torino Caselle: throughput simulazione 9

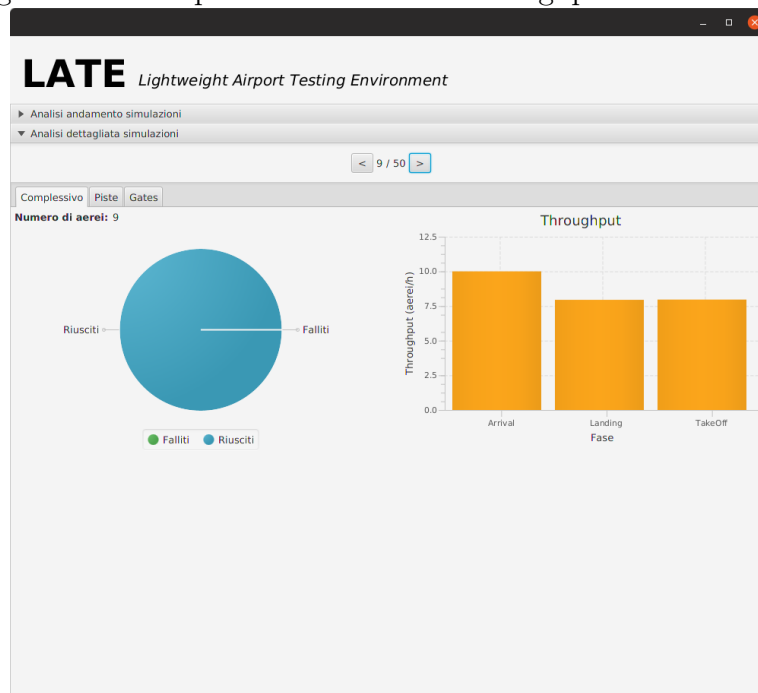


Figura 16: Esempio Torino Caselle: throughput simulazione 45

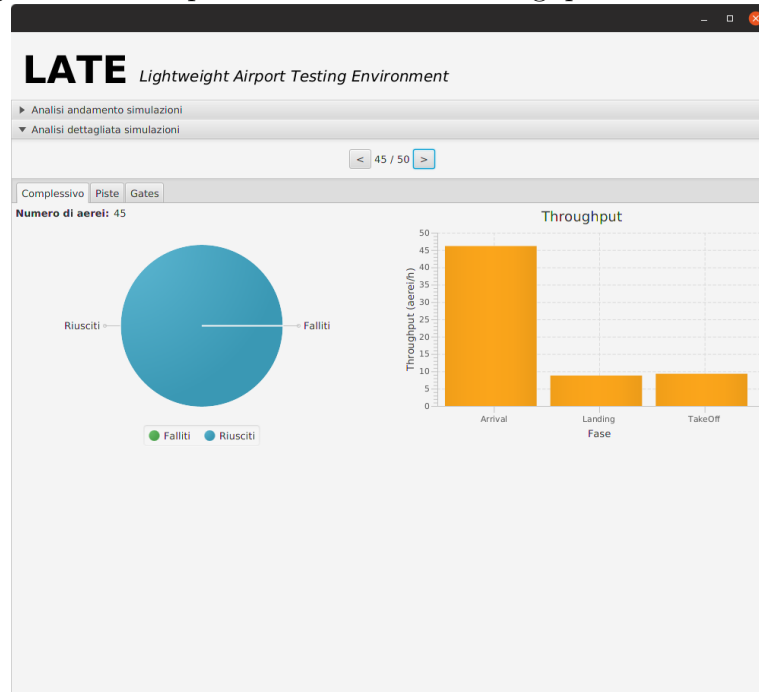
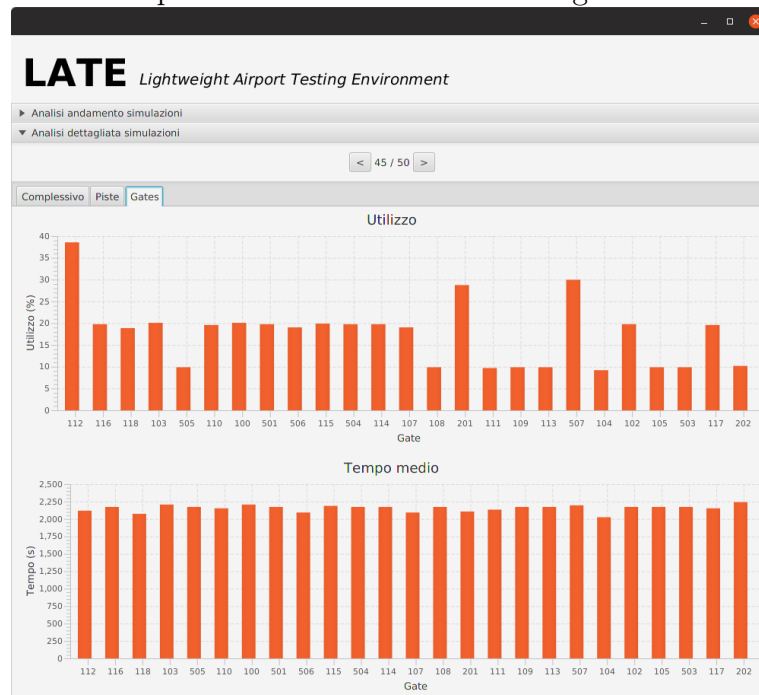


Figura 17: Esempio Torino Caselle: situazione gates simulazione 45



8 Conclusioni e possibili sviluppi

In conclusione, mi ritengo soddisfatto della riuscita del progetto che valuto rispondente agli obiettivi che mi ero prefissato.

Naturalmente le funzionalità della simulazione sono ampliabili e qui di seguito propongo quali, a mio avviso, potrebbero essere alcune possibilità di sviluppo:

- trasformerei il modello simulativo in una libreria, creando interfacce e classi astratte. Questo permetterebbe più agevolmente a collaboratori esterni di aggiungere funzionalità o di personalizzare quelle già esistenti, in pieno stile open-source;
- migliorerei il processo di avvicinamento alla pista che, come si è visto dall'esempio sopra riportato, è il bottleneck della simulazione. Esistono metodologie sofisticate, adottate negli aeroporti più trafficati, che permettono di aumentarne il throughput;
- introdurrei i ground services, che aggiungerebbero un livello di definizione in più all'analisi dei processi di turnaround degli aerei.

Tuttavia, ritengo il progetto valido, infatti:

- è veloce e leggero: riesce a eseguire un gran numero di simulazioni in poco tempo (in media 5 *ms* per simulazione);
- è user friendly: l'esperienza per gli utenti è semplificata dalla presentazione dei risultati in forma grafica, molto intuitiva;
- è funzionale: fornisce informazioni dettagliate sulla complessità dei processi che regolano l'attraversamento di un'infrastruttura aeroportuale.

Riferimenti bibliografici

- [1] Markets and Markets, “Air traffic management market by airspace (air traffic services, flow management, airspace management, aeronautical info management), component (hardware, software), end use (commercial, tactical), investment type, and region - global forecast to 2025,” 2018. [Online]. Available: <https://www.marketsandmarkets.com/Market-Reports/air-traffic-management-market-160955838.html>
- [2] B. J. Kim, A. A. Trani, X. Gu, and C. Zhong, “Computer simulation model for airplane landing-performance prediction,” *Transportation Research Record: Journal of the Transportation Research Board*, vol. 1562, no. 1, pp. 53–62, Jan. 1996. [Online]. Available: <https://pdfs.semanticscholar.org/e747/fa411dee8a76223cf47db6b4e69a4c9e2d1f.pdf>
- [3] W. users, “Avgas,” Jun 2020. [Online]. Available: <https://en.wikipedia.org/wiki/Avgas>
- [4] G. Austerman, “How large aircraft fuel up,” May 1997. [Online]. Available: <https://www.petroplaza.com/knowledge/1796>
- [5] E. W. Dijkstra, “A note on two problems in connexion with graphs,” *Numerische Mathematik*, vol. 1, no. 1, pp. 269–271, Dec. 1959. [Online]. Available: <https://doi.org/10.1007/bf01386390>

LATE (Lightweight Airport Testing Environment) by Augusto Colongo is licensed under CC BY-NC-SA 4.0. To view a copy of this license, visit <https://creativecommons.org/licenses/by-nc-sa/4.0>.

