

Homeworks: Architetture Soluzioni

Roberta Gori, Laura Semini
Ingegneria del Software
Dipartimento di Informatica
Università di Pisa

- Soluzioni degli esercizi risolti durante l'esercitazione
- Se mandate le soluzioni (.vpp) degli altri, saranno corrette e a fine corso pubblicate

Ex 1: REBU

Si assuma che sia stata definita un'architettura con i seguenti componenti:

- app_cliente (che implementa l'app usata dal cliente)
- sistema_prenotazione (che implementa la business logic del sistema)
- db_anagrafica (che raccoglie dati anagrafici su autisti e clienti)
- db_corse (che raccoglie dati su prenotazioni, corse assegnate, corse effettuate)
- app_autista (che implementa l'app usata dall'autista)

Si disegni il diagramma C&C, specificando gli stereotipi corrispondenti al tipo di interazione che si immagina sui connettori.

Ex2: REBU

Sotto le stesse assunzioni dell'esercizio 1, e assumendo un artefatto (omonimo) per componente, si disegni un diagramma di dislocazione (deployment) per il sistema.

Si assuma, che il db_anagrafica venga aggiornato poco frequentemente (solo quando si iscrivono nuovi autisti e nuovi clienti), e che venga interrogato con query semplici, mentre il db_corse sia soggetto ad aggiornamenti più frequenti, e debba eseguire query complesse (per esempio, per verificare quali autisti saranno liberi a un dato orario occorre incrociare le ore di inizio e fine servizio di tutte le corse attualmente prenotate e delle altre richieste pendenti).

REBU: Estensione Aeroporti

Il servizio di prenotazione di REBU si arricchisce di una nuova feature: la possibilità di associare un numero di volo, in caso di corse da/per un aeroporto.

Nel caso di spostamenti verso un aeroporto, al momento della richiesta, il cliente specifica il numero del volo e con quanto anticipo, in minuti, desidera arrivare in aeroporto rispetto alla partenza del volo. Il sistema si interfaccia con i servizi dell'aeroporto e ottiene l'orario di partenza. Si interfaccia quindi con un servizio di calcolo di percorso stradale (tipo google maps) per ottenere i tempi di percorrenza tra la posizione di ritrovo e l'aeroporto, e stabilire un orario di partenza per la corsa. A questo punto procede come con una normale prenotazione.

Nel caso di spostamenti da un aeroporto, al momento della richiesta, il cliente specifica il numero del volo e se avrà del bagaglio imbarcato da attendere dopo l'atterraggio. Il sistema si interfaccia con i servizi dell'aeroporto, ottiene l'orario di atterraggio del volo, stabilisce un orario di partenza per la corsa, e procede come con una normale prenotazione.

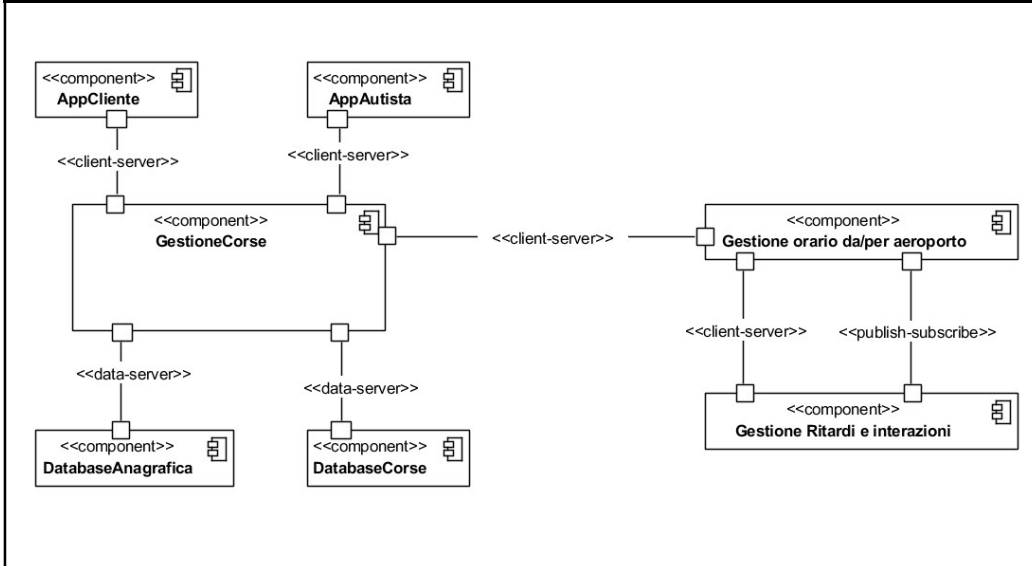
Nelle ultime due ore prima dell'inizio della corsa, REBU monitora eventuali ritardi dei voli. In caso di ritardo superiore ai 15 minuti, ricalcola l'orario di partenza della corsa, cerca un nuovo autista per la corsa ritardata (allo stesso prezzo della corsa già pattuita con l'utente), e cancella la precedente. Quindi informa l'utente del nuovo orario.

Ex3: REBU (Estensione Aeroporti)

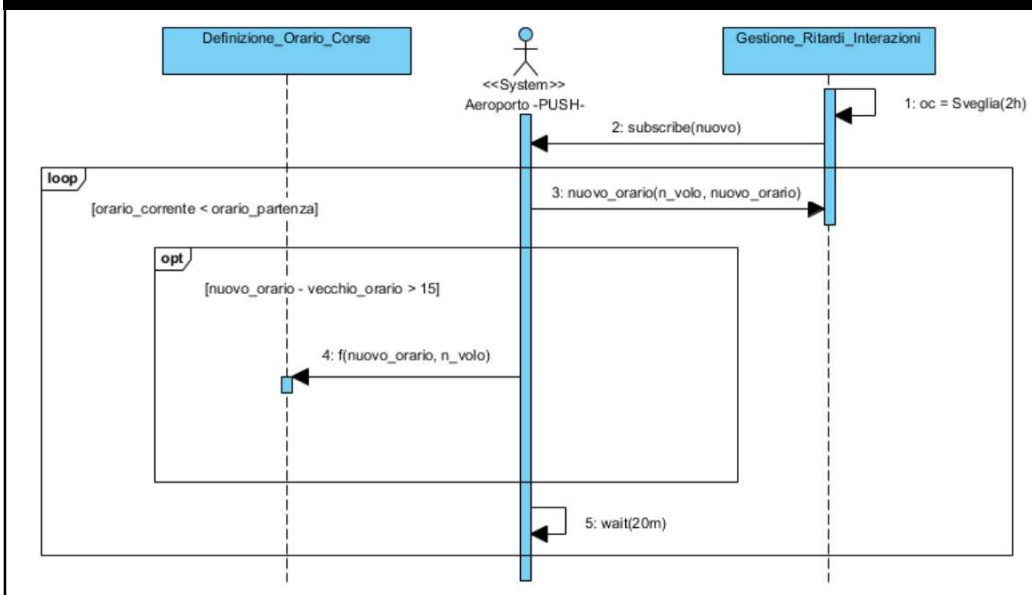
La progettazione architetturale ha portato all'individuazione di una componente GestioneRitardi che si occupa di monitorare eventuali ritardi dei voli e, se necessario, modificare la corsa. I servizi offerti dai vari aeroporti non sono uniformi: alcuni offrono la possibilità di abbonarsi alle informazioni su un volo e informano REBU di eventuali ritardi (modalità PUSH), altri offrono solo un servizio di informazioni domanda/risposta (modalità PULL).

Descrivere, con un diagramma di sequenza, le interazioni tra GestioneRitardi e l'aeroporto, nei due casi.

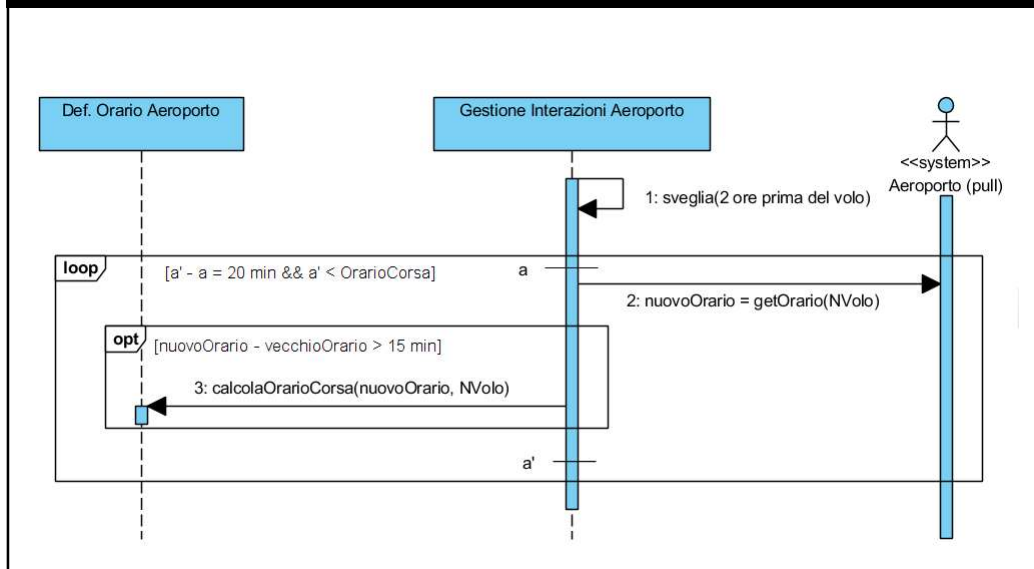
Es 3



Es 3 Push



Es 3 Pull



REBU: Estensione per condivisione auto

- Al fine di estendere il servizio, REBU attiva un programma di condivisione delle auto. Un autista durante le ore di riposo può decidere di lasciare la sua auto a disposizione di autisti che possono lavorare ma non hanno un'auto di proprietà con gli standard richiesti da REBU. A tal fine deve indicare la posizione dell'auto e l'intervallo orario in cui non la userà.
- Un autista che voglia usare un'auto condivisa cerca tra quelle disponibili, la prenota per il tempo necessario a raggiungerla (le prenotazioni durano max 20 minuti), ne prende possesso, lo segnala. Quando termina il turno la parcheggia in un raggio di 500 metri rispetto a dove l'ha trovata e segnala la nuova posizione. Il sistema provvede ad addebitare/accreditare il noleggio sull'account degli autisti coinvolti.
- Il meccanismo di apertura e messa in moto di auto condivise si basa sull'uso di un codice monouso inviato in risposta a una prenotazione.

Ex4: REBU (Estensione per condivisione auto)

La progettazione architetturale ha individuato le seguenti componenti:

- **DB auto**, che mantiene il parco auto, con il loro stato corrente,
- **DB autisti**, che mantiene informazioni sugli autisti, comprensive del loro conto economico;
- **Gestione Prestiti** che realizza la business logic.

Dare

- La vista C&C
- un diagramma di sequenza che mostri come queste componenti e gli attori coinvolti realizzano il caso d'uso Presa in Prestito, dal momento in cui un autista inizia una ricerca fra le auto condivise, fino a quando segnala la presa di possesso di un'auto. Il caso d'uso prevede una notifica al proprietario nel momento in cui l'auto è prenotata, e un'altra quando viene effettivamente presa in uso.

Ex5 ed Ex 6: PisaMover

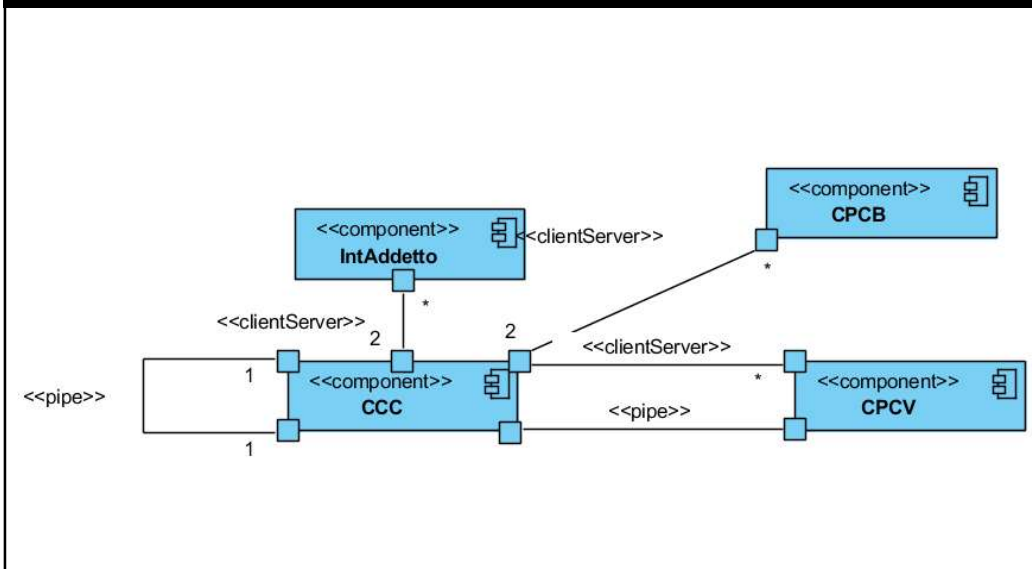
Si consideri il sottosistema che controlla il movimento e l'apertura-chiusura delle porte del PisaMover, che abbreviamo in SSC. Per garantirne la fault-tolerance, è stata adottata una politica di ridondanza: i server fisici sono duplicati e così le componenti software. In caso di fallimenti di un server, la componente sul server replica prende il controllo. In particolare, sono state individuate le seguenti componenti e le seguenti risorse hardware su cui dislocare gli artefatti che manifestano le componenti date.

Componente	Specifica	HW
IntAddetto	Interfaccia utente, per impostare la modalità operativa e inviare altri comandi alla CCC	PC, presso la sala controllo
Componente centrale di controllo (CCC)	Decide apertura/chiusura porte e movimento/arresto. Comunica le decisioni alle componenti periferiche CPCB e CPCV	Server, presso la sala controllo
Componente periferica di controllo porte banchina (CPCB)	Riceve ordini dalla CCC, invia agli attuatori il segnale apertura/chiusura porte	Microcontrollore embedded, presso ogni stazione
Componente periferica di controllo vagoni (CPCV)	Riceve ordini dalla CCC, invia agli attuatori il segnale apertura/chiusura porte, trasmette alla CCC le richieste di passare in modalità Emergenza 2	Microcontrollore embedded, su ogni vagone

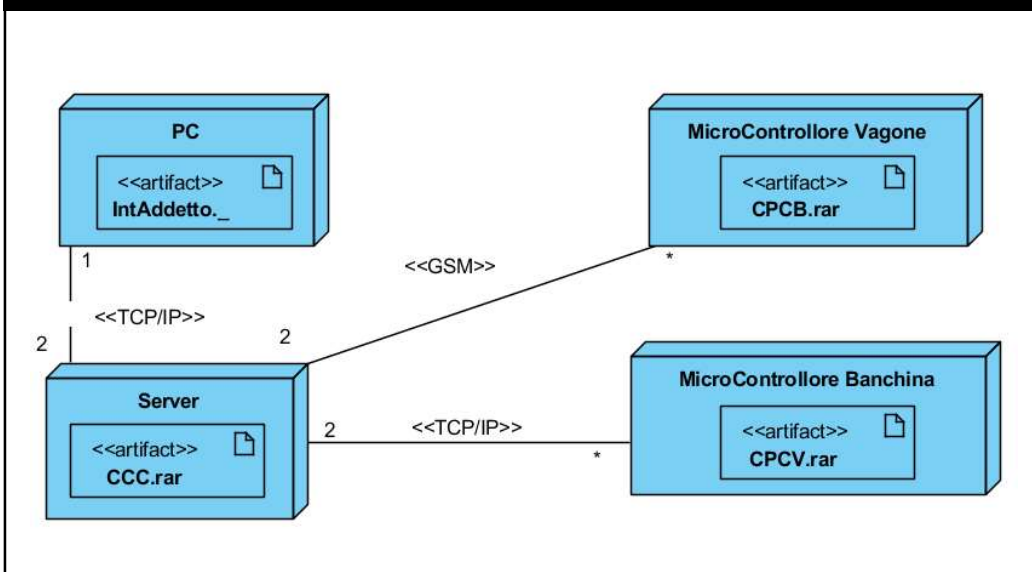
Ex5: Dare la vista Componenti & Connettori del SSC.

Ex6: Dare la vista di dislocazione del SSC, sapendo che le stazioni sono collegate tra loro con una rete fissa (su tecnologia TCP/IP), e che le macchine localizzate sui vagoni comunicano via rete dati GSM con la sala controllo.

Es 5



Es 6



Ex 8: PisaMover (Estensione Parcheggio)

Dopo un colloquio con i committenti, sono stati meglio specificati i termini per l'uso del parcheggio scambiatore. Quando l'autista arriva alla sbarra, trova una macchinetta dove acquista i biglietti, indicando il numero di persone. La macchina restituisce un biglietto per auto+conducente e un biglietto per ogni passeggero. I biglietti permettono alle persone di superare i tornelli e accedere ai binari. Prima di ritirare l'auto, se il parcheggio è durato più di 18 ore, il conducente deve pagare il supplemento presso una macchinetta posta in prossimità dei binari, che aggiorna i dati sul biglietto auto+conducente, da quel momento il conducente ha 5 minuti per uscire dal parcheggio. I 5 minuti di comporto sono riconosciuti anche a chi non deve pagare alcun supplemento. Per uscire, il conducente deve inserire il biglietto auto+conducente in un lettore prossimo alla sbarra di uscita.

Ex 8: Il sottosistema dei parcheggi prevede numerosi componenti attivi: emettitrici di biglietti, tornelli, macchine per il pagamento del supplemento orario, sbarre, ecc. Assumendo che ciascuna di questi abbia un microcontrollore locale, che ne controlla il funzionamento, si fornisca una vista ibrida UML (deployment e C&C) che illustri come si intenderebbe distribuire il software necessario al funzionamento del sistema.

Ex 9: PisaMover

Si consideri il sottosistema di Pisa Mover, chiamato sistema di controllo marcia (SCM). Questa è la parte del sistema che si occupa di comandare i motori che, tramite funi di traino (cavi di acciaio), controllano il movimento dei vagoni, e di controllare la posizione dei vagoni lungo i binari.

Ex9: Si è deciso di progettare il SCM installando numerosi sensori di presenza dei vagoni lungo il binario e alle stazioni: almeno uno per ogni zona e uno al confine tra le zone. Inoltre, si è deciso di realizzare l'SCM sotto forma di due moduli identici (ovvero, due istanze dello stesso componente), ciascuno dei quali controlla i movimenti di uno dei due convogli. Si dia un diagramma di sequenza che mostri le comunicazioni fra uno di tali moduli, sensori e motori, nel corso di un viaggio di un vagone da un capo all'altro del tracciato.

Ex 10: PisaMover

Con riferimento al caso di studio Pisa Mover si consideri la nuova tariffa scontata per i biglietti stazione ferroviaria-aeroporto: dal primo di dicembre 2017 è possibile, per ogni residente del comune di Pisa, acquistare fino a 6 biglietti al mese al costo di 1,20€ cadauno anziché 2,70€. Al momento l'acquisto è possibile solo presso la biglietteria aziendale situata nel parcheggio Via Aurelia, presentando la carta di identità, ma si desidera modificare le biglietterie automatiche in modo da permettere la vendita di biglietti scontati. L'utente interessato deve registrarsi presso la biglietteria aziendale o via web, fornendo generalità, comune di residenza e codice fiscale. Il sistema controlla la veridicità della dichiarazione usando i dati mantenuti in locale, e in caso di verifica positiva il suo codice fiscale viene inserito tra quelli accettati dalle biglietterie automatiche. In questo modo l'utente potrà acquistare biglietti a prezzo scontato presso le biglietterie automatiche, indicando il codice fiscale. Ogni tre mesi il sistema richiede la lista dei residenti del comune di Pisa, con una richiesta a un servizio offerto dall'Anagrafe, e aggiorna il proprio database: elenco dei residenti da consultare per future richieste di registrazione; lista dei codici fiscali accettati dalle biglietterie automatiche, (rimuovendo i non più residenti).

Ex 10: Si disegni un diagramma UML ibrido, C&C/deployment, in cui si evidenziano i principali componenti software e hardware necessari per realizzare il nuovo requisito relativo alla gestione dei biglietti scontati.