AcmeBikes

Studente: Michele Abruzzese
Matricola: 0001097676

Studente: Luca Castricini
Matricola: 0001102690

Studente: Federico Augelli
Matricola: 0001097518

Progetto di Ingegneria del software orientata ai servizi
18 febbraio 2024

Prof:
Prof. Davide Rossi
Prof. Ivan Lanese

Alma Mater Studiorum
Bologna

Introduzione

Il presente rapporto illustra il processo di progettazione e realizzazione di un sistema informativo per l'azienda ACMEbikes, specializzata nella fornitura di biciclette assemblate su richiesta e accessori a rivendite che operano direttamente con il pubblico. L'obiettivo principale è stato quello di sviluppare un'architettura orientata ai servizi (SOA) che supportasse efficacemente le attività di ACMEbikes, garantendo un flusso efficiente di comunicazione e un'adeguata gestione dei processi aziendali.

Nel capitolo 2, "Dominio del problema", viene fornita una dettagliata descrizione delle attività svolte da ACMEbikes e dei requisiti funzionali del sistema. Tale analisi si basa sulla traccia progettuale fornita, che delinea chiaramente le operazioni coinvolte nel processo di vendita e fornitura di biciclette e accessori. Questa sezione costituisce il punto di partenza per la comprensione approfondita del contesto operativo dell'azienda.

Nel capitolo 3, "Comunicazione", viene presentata la comunicazione tra gli attori coinvolti nel processo tramite un diagramma di coreografia, raffinato iterativamente per migliorare le proprietà di connectedness. Tale rappresentazione grafica fornisce una panoramica chiara delle interazioni tra ACMEbikes, le rivendite, i corrieri, il sistema bancario e altri soggetti esterni. Inoltre, viene proiettata questa coreografia in un sistema di ruoli per una comprensione più dettagliata delle dinamiche di comunicazione.

Nel capitolo 4, "Modellazione del processo", viene presentato un diagramma di collaborazione BPMN che illustra l'intero processo aziendale, inclusi i dettagli di ciascun partecipante. Questo diagramma, pensato a scopo documentativo, fornisce una rappresentazione chiara e strutturata delle attività svolte da ACMEbikes e dalle rivendite, consentendo una comprensione dettagliata delle operazioni aziendali.

Nel capitolo 5, "Diagramma UML", si illustra il diagramma UML che descrive la SOA utilizzando i profili TinySOA.

Nel capitolo 6, "Sviluppo", si descrive lo sviluppo del sistema analizzando brevemente il funzionamento di ciascun componente.

Indice

Capitolo 1

Dominio del problema

1.0.1 Specifiche

L'azienda ACMEbikes si occupa di fornire biciclette assemblate su richiesta ed accessori a rivendite che operano direttamente con il pubblico.

Le rivendite inviano ad ACMEbikes ordini composti da cicli e accessori. Per ogni ciclo viene specificato il modello base, la colorazione e le eventuali customizzazioni per gruppo frenante, gruppo sterzo, trasmissione e ammortizzatori. Una volta ricevuto l'ordine ACMEbikes si accerta che le eventuali customizzazioni richieste siano possibili e procede quindi a presentare un preventivo alla rivendita (vedi sotto), se il preventivo viene accettato ACMEbikes procede all'assemblaggio dei cicli richiesti e alla fornitura degli accessori.

L'assemblaggio prevede una prima fase nella quale si verifica la presenza di tutti componenti necessari nel magazzino principale; nel caso di componenti non presenti si verifica se sono in uno dei magazzini secondari. Se ci sono vengono trasferiti, se non ci sono vengono ordinati (ad un unico fornitore esterno).L'assemblaggio dei cicli avviene nella sede principale dell'azienda, dove si trova anche il magazzino principale (ma si sta valutando se spostarlo in una nuova sede).

Per maggiore efficienza, se un ordine include accessori che non devono essere assemblati (o che richiedono un assemblaggio elementare, come ad esempio un fanale), questo può essere scomposto in modo che sia possibile che i diversi prodotti vengano inviati al cliente direttamente dai magazzini. A tal fine, per ogni prodotto, il magazzino che viene selezionato è quello che ha il prodotto disponibile e che è geograficamente più vicino alla sede del cliente. Tutte le spedizioni verso i clienti vengono realizzate tramite un servizio di corriere esterno, le spedizioni fra le sedi di ACMEbikes avvengono con i mezzi dell'azienda.

Il costo dell'ordine viene determinato in funzione del costo dei prodotti e di tutte le spedizioni necessarie. Per ordini superiori ad una certa cifra (che può variare nel tempo) viene valutata l'e-

ventuale applicazione di uno sconto. Tale valutazione viene effettuata da un membro dello staff di ACMEbikes che ha l'autorità per farla. Una volta determinato il costo finale viene inviato un preventivo al cliente che può accettarlo o rifiutarlo. In caso di rifiuto l'interazione termina.

In caso di accettazione ACMEbikes si pone in attesa che il cliente invii un riferimento ad un bonifico effettuato per una somma pari almeno ad un decimo del costo dell'ordine a titolo di anticipo. Una volta verificato con il sistema bancario il pagamento dell'anticipo, ACMEbikes procede alle spedizioni e si pone in attesa che il cliente notifichi il pagamento del saldo. Una volta controllato con il sistema bancario anche questo trasferimento il processo termina.

1.0.2 Workflow e artefatti

Si progetti e si realizzi una SOA che supporti le attività di ACMEbikes. Si modellino le comunicazioni dello scenario sopra esposto usando una coreografia, si discutano le sue proprietà di connectedness ed eventualmente si raffini la coreografia per migliorare tali proprietà. Si proietti la coreografia in un sistema di ruoli.

Utilizzando uno o più diagrammi di collaborazione BPMN si modelli l'intera realtà descritta compresi i dettagli di ogni partecipante riferibile ad ACMEbikes e delle rivendite. Tale modellazione ha scopo documentativo quindi il livello di dettaglio deve essere consistente con tale scopo. I partecipanti "esterni" (corriere, sistema bancario, ecc. . .) possono apparire come collapsed pools.

Si progetti una SOA per la realizzazione del sistema e la si documenti utilizzando UML (eventualmente con opportuni profili, ad esempio TinySOA).

Si realizzi il sistema usando come tecnologie un BPMS (si consiglia di utilizzare Camunda), Jolie e API Rest. Il BPMS deve essere utilizzato per supportare i processi di ACMEbikes e delle rivendite. La parte di gestione dei magazzini (principale e secondari) deve essere effettuata tramite una SOA di servizi Jolie. Si assume che il sistema integri sotto forma di servizi (almeno) le seguenti capability esterne:

- Calcolo distanze geografiche (preferibilmente con API Rest)
- Sistema bancario (preferibilmente API Rest)
- Fornitore componenti
- Corriere

Tali servizi vanno implementati (con logica elementare) come parte del progetto.

I modelli di processo BPMN da utilizzare per il BPMS devono essere consistenti con la modellazione a scopo documentativo precedentemente realizzata; volendo si può anche scegliere di dettagliare compiutamente già dal primo modello le pool eseguibili. Quindi nel primo caso si avrebbe un modello BPMN documentativo e poi tanti modelli BPMN eseguibili quanti i partecipanti realizzati attraverso BPMS; in alternativa si avrebbe un unico modello BPMN con le pool eseguibili completamente dettagliate e gli altri partecipanti dettagliati a livello documentativo. Il dialogo fra Jolie e BPMS deve avvenire via SOAP

1.1 Assunzioni e semplificazioni

- **Database condiviso**
  - Per agevolare la fase di verifica della customizzazione che comprende l'accertamento della presenza di tutti i pezzi nei vari magazzini dell'azienda, abbiamo deciso di utilizzare un'unica base dati nel quale AcmeBikes verifica la disponibilità dei pezzi nei vari magazzini, in modo tale da diminuire le comunicazioni tra quest'ultima e i magazzini. In questo modo AcmeBikes invierà una lista dei componenti necessari per l'assemblaggio direttamente ai magazzini che dispongono dei componenti.
- **Sistema bancario**
  - Il sistema bancario è stato implementato usando una logica semplificata: Il servizio che si occupa di gestire i pagamenti tra le rivendite e AcmeBikes crea un token alla ricezione del pagamento che verrà successivamente inviato ad AcmeBikes per verificare la correttezza dell'importo. Non sono previste logiche aggiuntive riguardo eventuali ricariche del conto corrente da parte della rivendita.
- **Fornitore esterno e Corriere**
  - Entrambi vengono implementati con logica elementare, ovvero:
    * Quando il fornitore esterno riceve un ordine dai magazzini risponde con un messaggio che indica l'invio dei pezzi necessari (non è contemplata l'opzione per il quale il fornitore esterno non sia in grado di soddisfare l'ordine).
    * Quando il corriere esterno dialoga con i magazzini e AcmeBikes riceve un messaggio con una lista di componenti e automaticamente invia un messaggio alle rivendite che viene inteso come la consegna dei componenti. Viene inviato un solo messaggio da parte del corriere verso le rivendite in modo tale che le rivendite possano eseguire il pagamento per saldare il conto.
- **Sede magazzino principale**
  - Tenendo in considerazione, come da traccia, che si sta pensando di spostare il magazzino principale in una nuova sede, abbiamo già scorporato il magazzino principale dalla sede principale. Questo si può vedere dal diagramma BPMN generale che viene mostrato in

seguito. L'assemblaggio dei cicli avviene direttamente nella sede principale, quindi tutti i magazzini inviano i pezzi da assemblare alla sede principale.

Capitolo 2

Comunicazioni

Esploreremo le dinamiche comunicative del processo aziendale di ACMEbikes attraverso l'utilizzo di una coreografia. Questo strumento di modellazione ci consente di rappresentare in modo chiaro e integrato le interazioni tra ACMEbikes, le rivendite, i corrieri, il sistema bancario e altri soggetti esterni. La coreografia viene raffinata iterativamente per migliorare le proprietà di connectedness, garantendo una rappresentazione accurata delle comunicazioni. Successivamente, proiettiamo la coreografia in un sistema di ruoli per comprendere meglio le responsabilità di ciascun attore nel processo. In breve, l'utilizzo di una coreografia offre una visione completa delle comunicazioni aziendali, fornendo una base solida per la progettazione del sistema informativo

2.1 Coreografia

```
1   CoreografiaOrdine :== (
2       InvioOrdine : Rivendita -> AcmeBike;
3       (
4               CastomzzazioneNonFattibile : AcmeBike -> Rivendita
5       )+(
6               InvioPreventivo : AcmeBike -> Rivendita;
7               (
8                       //Timer
9               )+(
10                      AccettaPreventivo : Rivendita -> AcmeBikes;
11                      LoginBanca : Rivendita -> Banca;
12                      ConfermaLogin : Banca -> Rivendita;
13                      InvioPagamento : Rivendita -> Banca;
14                      RilasciaToken : Banca -> Rivendita;
15                      InvioToken : Rivendita -> AcmeBikes;
16                      LoginBanca : AcmeBike -> Banca;
17                      ConfermaLogin : Banca -> AcmeBikes;
18                      ConvalidaToken : AcmeBikes -> Banca;
19                      (
20                              TokenInvalid : Banca -> AcmeBikes;
21                              NotifyTokenInvalid : AcmeBike -> Rivendita;
22                      )+(
23                              TokenValid : Banca -> AcmeBikes;
24                              AccontoAccettato : AcmeBike -> Rivendita;
25                              (
26                                      InviaListaComponenti : AcmeBike -> MagazzinoPrin;
27                                      (
28                                              InviaListaCompEst : MagazzinoPrin -> FornitoreEst;
29                                              InviaComponenti : FornitoreEst -> MagazzinoPrin;
30                                      )+(
31                                              (
32                                                      InviaComponenti : MagazzinoPrin -> CorriereEst;
33                                                      InviaComponenti : CorriereEst -> Rivendita;
34                                              )|(
35                                                      InviaComponenti : MagazzinoPrin -> AcmeBikes;
36
37                                              )
38                                      )
39                              )|(
40                                      InviaListaComponenti : AcmeBike -> MagazzinoSec;
41                                      (
42                                              InviaListaCompEst : MagazzinoSec -> FornitoreEst;
43                                              InviaComponenti : FornitoreEst -> MagazzinoSec;
44
45                                              (
46                                                      InviaComponenti : MagazzinoSec -> CorriereEst;
47                                                      InviaComponenti : CorriereEst -> Rivendita;
48                                              )|(
49                                                      InviaComponenti : MagazzinoSec -> AcmeBikes;
50
51                                              )
52                                      )
53                              )
54                              InviaCicli : AcmeBikes -> CorriereEst;
55                              InviaComponenti : CorriereEst -> Rivendita;
56                              LoginBanca : Rivendita -> Banca;
57                              ConfermaLogin : Banca -> Rivendita;
58                              InvioPagamento : Rivendita -> Banca;
59                              RilasciaToken : Banca -> Rivendita;
60                              InvioToken : Rivendita -> AcmeBikes;
61                              LoginBanca : AcmeBikes -> Banca;
62                              ConfermaLogin: Banca -> AcmeBikes;
63                              ConvalidaToken : AcmeBikes -> Banca;
64                              (
65                                      TokenValid : Banca -> AcmeBikes;
66                                      NotifyTokenValid : AcmeBike -> Rivendita;
67                              )+(
68                                      TokenInvalid : Banca -> AcmeBikes;
69                                      NotifyTokenInvalid : AcmeBike -> Rivendita;
70                                      LoginBanca : Rivendita -> Banca;
71                              )
72
73                      )
74              )
75      )
76  )
```

2.2 Connectedness

Durante la fase iniziale di lavoro, la coreografia è stata sottoposta a diversi raffinamenti al fine di migliorare le sue tre proprietà di connectedness: composizione sequenziale, scelta ed uso multipli della stessa operazione. Questo processo è stato finalizzato a garantire che la coreografia funzioni correttamente. Per migliorare la correttezza della composizione sequenziale, sono stati aggiunti ACK dove possibile. Per quanto riguarda la correttezza dei punti di scelta, sono state effettuate verifiche per assicurare che lo stesso ruolo compaia in ogni transizione iniziale per ogni punto di scelta e che i ruoli relativi ai branches siano gli stessi. La correttezza delle iterazioni è ottenuta rispettando i punti precedenti, in quanto possono essere viste come una combinazione di composizioni sequenziali e punti di scelta.

- Alla riga due AcmeBike è destinatario in InvioOrdine ed è quindi mittente sia in CastomzzazioneNonFattibile che in InvioPreventivo;
- Alla riga 6 in InvioPreventivo, Rivendite è destinatario ed è quindi mittente in AccettaPreventivo;
- Alla riga 18 Banca è destinatario di ConvalidaToken, diventa quindi mittente in TokenInvalid e in TokenValid
- Alla riga 26 MagagginoPrin è destinatario di InviaListaComponenti, diventa quindi mittente in InviaListaCompEst e in InviaComponenti
- Alla riga 40 MagazzinoSec è destinatario di InviaListaComponenti ed è quindi mittente in InviaListaCompEst e in InviaComponenti
- Alla riga 63 Banca è destinatario di ConvalidaToken, diventa quindi mittente in TokenValid e TokenInvalid

2.3 Sistema di ruoli

I ruoli individuati sono i seguenti: AcmeBikes, Banca, CorriereEst, FornitoreEst,MagazzinoPrin, MagazzinoSec, Rivendite. La struttura di ogni ruolo è la stessa della coreografia in 3.1 in modo tale da avere corrispondenza nei numeri di riga.

2.3.1 AcmeBikes

```
CoreografiaOrdine :== (
        InvioOrdine@Rivendita;
        (
                CastomzzazioneNonFattibile@Rivendita
        )+(
                InvioPreventivo@Rivendita;
                (
                        //Timer
                )+(
                        AccettaPreventivo@Rivendita;
                        1 ;
                        1 ;
                        1 ;
                        1 ;
                        InvioToken@Rivendita;
                        LoginBanca@Banca;
                        ConfermaLogin@Banca;
                        overlineConvalidaToken@Banca;
                        (
                                TokenInvalid@Banca;
                                NotifyTokenInvalid@Rivendita;
                        )+(
                                TokenValid@Banca;
                                AccontoAccettato@Rivendita;
                                (
                                        InviaListaComponenti@MagazzinoPrin;
                                        (
                                                ;
                                                ;
                                        )+(
                                                (
                                                        ;
                                                        ;
                                                )|(
                                                        InviaComponenti@MagazzinoPrin;

                                                )
                                        )
                                )|(
                                        InviaListaComponenti@MagazzinoSec;
                                        (
                                                ;
                                                ;

                                                (
                                                        ;
                                                        ;
                                                )|(
                                                        InviaComponenti@MazzinoSec;

                                                )
                                        )
                                )
                                InviaCicli@CorriereEst;
                                ;
                                ;
                                ;
                                ;
                                ;
                                InvioToken@Rivendita;
                                LoginBanca@AcmeBikes;
                                ConfermaLogin@Banca;
                                ConvalidaToken@Banac;
                                (
                                        TokenValid@Banca;
                                        NotifyTokenValid@Rivendita;
                                )+(
                                        TokenInvalid@Banca;
                                        NotifyTokenInvalid@Rivendita;
                                        LoginBanca@Rivendita;
                                )

                        )
                )
        )
)
```

2.3.2 Banca

```
CoreografiaOrdine :== (
        1;
        (
                1
        )+(
                1;
                (
                        //Timer
                )+(
                        1;
                        LoginBanca@Rivendita;
                        ConfermaLogin@Rivendita;
                        InvioPagamento@Rivendita;
                        RilasciaToken@Rivendita;
                        1;
                        LoginBanca@AcmeBike;
                        ConfermaLogin@AcmeBikes;
                        ConvalidaToken@AcmeBikes;
                        (
                                TokenInvalid@AcmeBikes;
                                1;
                        )+(
                                TokenValid@AcmeBikes;
                                1;
                                (
                                        1;
                                        (
                                                1;
                                                1;
                                        )+(
                                                (
                                                        1;
                                                        1;
                                                )|(
                                                        1;

                                                )
                                        )
                                )|(
                                        1;
                                        (
                                                1;
                                                1;

                                                (
                                                        1;
                                                        1;
                                                )|(
                                                        1;

                                                )
                                        )
                                )
                                1;
                                1;
                                LoginBanca@Rivendita;
                                ConfermaLogin@Rivendita;
                                InvioPagamento@Rivendita;
                                RilasciaToken@Rivendita;
                                1;
                                LoginBanca@AcmeBikes;
                                ConfermaLogin@AcmeBikes;
                                ConvalidaToken@AcmeBikes;
                                (
                                        TokenValid@AcmeBikes;
                                        1;
                                )+(
                                        TokenInvalid@AcmeBikes;
                                        1;
                                        LoginBanca@Rivendita;
                                )

                        )
                )
        )
)
```

2.3.3 Corriere esterno

```
CoreografiaOrdine :== (
        1;
        (
                1
        )+(
                1;
                (
                        //Timer
                )+(
                        1;
                        1;
                        1;
                        1;
                        1;
                        1;
                        1;
                        1;
                        1;
                        (
                                1;
                                1;
                        )+(
                                1;
                                1;
                                (
                                        1;
                                        (
                                                1;
                                                1;
                                        )+(
                                                (
                                                        InviaComponenti@MagazzinoPrin;
                                                        InviaComponenti@Rivendita;
                                                )|(
                                                        1;

                                                )
                                        )
                                )|(
                                        1;
                                        (
                                                1;
                                                1;

                                                (
                                                        InviaComponenti@MagazzinoSec;
                                                        InviaComponenti@Rivendita;
                                                )|(
                                                        1;

                                                )
                                        )
                                )
                                InviaCicli@AcmeBikes;
                                InviaComponenti@Rivendita;
                                1;
                                1;
                                1;
                                1;
                                1;
                                1;
                                1;
                                1;
                                (
                                        1;
                                        1;
                                )+(
                                        1;
                                        1;
                                        1;
                                )

                        )
                )
        )
)
```

2.3.4 Fornitore esterno

```
CoreografiaOrdine :== (
        1;
        (
                1
        )+(
                1;
                (
                        //Timer
                )+(
                        1;
                        1;
                        1;
                        1;
                        1;
                        1;
                        1;
                        1;
                        1;
                        (
                                1;
                                1;
                        )+(
                                1;
                                1;
                                (
                                        1;
                                        (
                                                InviaListaCompEst@MagazzinoPrin;
                                                InviaComponenti@MagazzinoPrin;
                                        )+(
                                                (
                                                        1;
                                                        1;
                                                )|(
                                                        1;

                                                )
                                        )
                                )|(
                                        1;
                                        (
                                                InviaListaCompEst@MagazzinoSec;
                                                InviaComponenti@MagazzinoSec;

                                                (
                                                        1;
                                                        1;
                                                )|(
                                                        1;

                                                )
                                        )
                                )
                                1;
                                1;
                                1;
                                1;
                                1;
                                1;
                                1;
                                1;
                                1;
                                (
                                        1;
                                        1;
                                )+(
                                        1;
                                        1;
                                        1;
                                )

                        )
                )
        )
)
```

2.3.5 Magazzino principale

```
CoreografiaOrdine :== (
        1;
        (
                1
        )+(
                1;
                (
                        //Timer
                )+(
                        1;
                        1;
                        1;
                        1;
                        1;
                        1;
                        1;
                        1;
                        1;
                        (
                                1;
                                1;
                        )+(
                                1;
                                1;
                                (
                                        InviaListaComponenti@AcmeBike;
                                        (
                                                InviaListaCompEst@FornitoreEst;
                                                InviaComponenti@FornitoreEst;
                                        )+(
                                                (
                                                        InviaComponenti@CorriereEst;
                                                        1;
                                                )|(
                                                        InviaComponenti@AcmeBikes;

                                                )
                                        )
                                )|(
                                        1;
                                        (
                                                1;
                                                1;

                                                (
                                                        1;
                                                        1;
                                                )|(
                                                        1;

                                                )
                                        )
                                )
                                1;
                                1;
                                1;
                                1;
                                1;
                                1;
                                1;
                                1;
                                1;
                                1;
                                (
                                        1;
                                        1;
                                )+(
                                        1;
                                        1;
                                        1;
                                )

                        )
                )
        )
)
```

2.3.6 Magazzino secondario

```
CoreografiaOrdine :== (
        1;
        (
                1
        )+(
                1;
                (
                        //Timer
                )+(
                        1;
                        1;
                        1;
                        1;
                        1;
                        1;
                        1;
                        1;
                        1;
                        (
                                1;
                                1;
                        )+(
                                1;
                                1;
                                (
                                        1;
                                        (
                                                1;
                                                1;
                                        )+(
                                                (
                                                        1;
                                                        1;
                                                )|(
                                                        1;

                                                )
                                        )
                                )|(
                                        InviaListaComponenti@AcmeBike;
                                        (
                                                InviaListaCompEst@FornitoreEst;
                                                InviaComponenti@FornitoreEst;

                                                (
                                                        InviaComponenti@CorriereEst;
                                                        1;
                                                )|(
                                                        InviaComponenti@AcmeBikes;

                                                )
                                        )
                                )
                                1;
                                1;
                                1;
                                1;
                                1;
                                1;
                                1;
                                1;
                                1;
                                (
                                        1;
                                        1;
                                )+(
                                        1;
                                        1;
                                        1;
                                )

                        )
                )
        )
)
```

2.3.7 Rivendite

```
CoreografiaOrdine :== (
        InvioOrdine@AcmeBike;
        (
                CastomzzazioneNonFattibile@AcmeBike
        )+(
                InvioPreventivo@AcmeBike;
                (
                        //Timer
                )+(
                        AccettaPreventivo@AcmeBikes;
                        LoginBanca@Banca;
                        ConfermaLogin@Banca;
                        InvioPagamento@Banca;
                        RilasciaToken@Banca;
                        InvioToken@AcmeBikes;
                        1;
                        1;
                        1;
                        (
                                1;
                                NotifyTokenInvalid@AcmeBike;
                        )+(
                                1;
                                AccontoAccettato@AcmeBike;
                                (
                                        1;
                                        (
                                                1;
                                                1;
                                        )+(
                                                (
                                                        1;
                                                        InviaComponenti@CorriereEst;
                                                )|(
                                                        1;

                                                )
                                        )
                                )|(
                                        1;
                                        (
                                                1;
                                                1;

                                                (
                                                        1;
                                                        InviaComponenti@CorriereEst;
                                                )|(
                                                        1;

                                                )
                                        )
                                )
                                1;
                                InviaComponenti@CorriereEst;
                                LoginBanca@Banca;
                                ConfermaLogin@Rivendita;
                                InvioPagamento@Banca;
                                RilasciaToken@Banca;
                                InvioToken@AcmeBikes;
                                1;
                                1;
                                1;
                                (
                                        1;
                                        NotifyTokenValid@AcmeBike;
                                )+(
                                        1;
                                        NotifyTokenInvalid@AcmeBike;
                                        LoginBanca@Banca;
                                )

                        )
                )
        )
)
```

2.4 Diagramma di coreografia BPMN

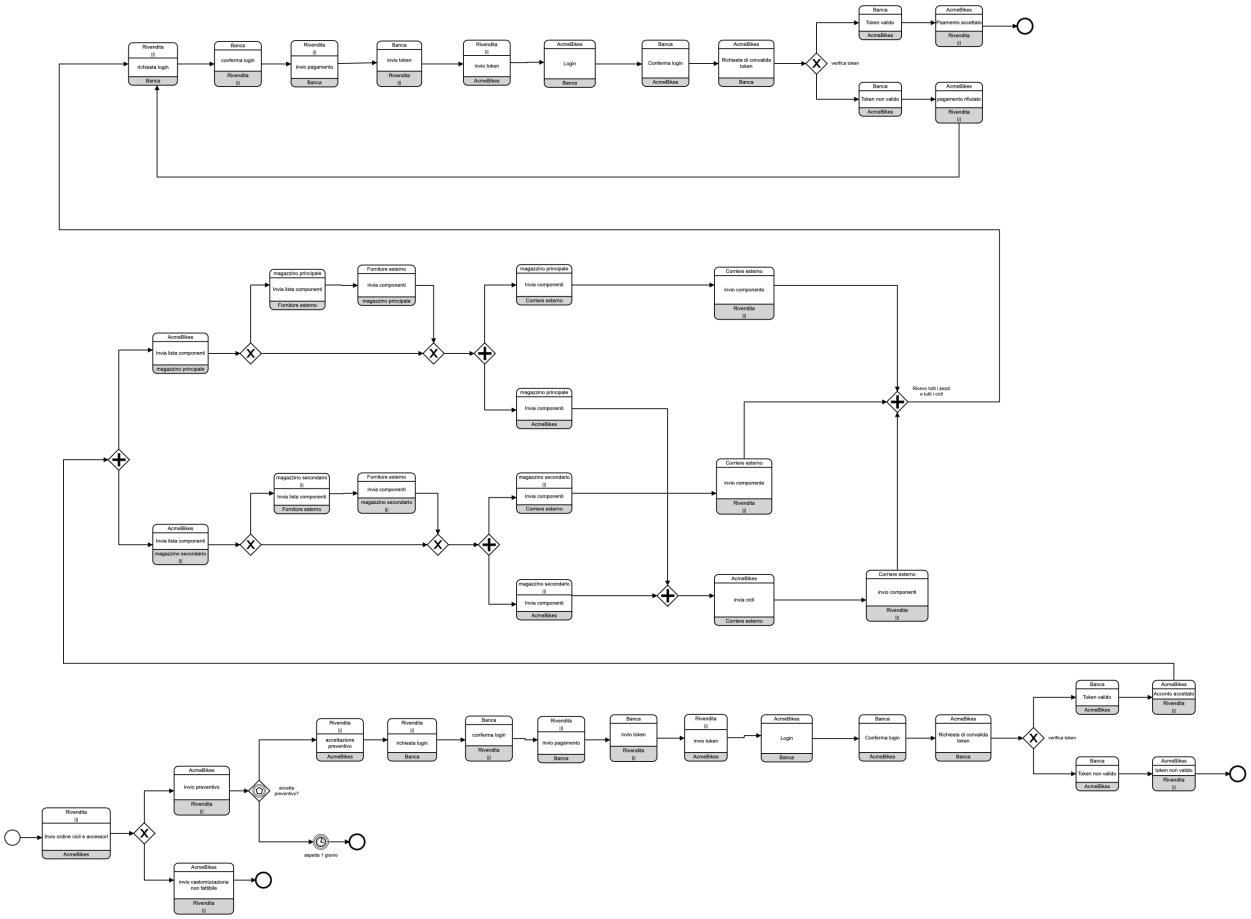

Figura 2.1: Diagramma di coreografia BPMN

Capitolo 3

Modellazione

A seguito della descrizione, abbiamo rappresentato la coreografia con un diagramma di coreografia BPMN, utile nelle fasi successive grazie alla maggiore leggibilità rispetto alla coreografia precedente.

È stato elaborato un diagramma di processo BPMN generale a scopo documentativo, il quale rappresenta la coreografia sviluppata in precedenza. Questo diagramma fornisce una panoramica completa delle interazioni e delle attività coinvolte nel processo aziendale di ACMEbikes. Successivamente, utilizzando il BPMS Camunda, sono stati realizzati due diagrammi eseguibili. Questi diagrammi sono stati progettati per simulare e automatizzare l'intero processo descritto nella traccia del progetto. Attraverso l'implementazione in Camunda, è stato possibile tradurre efficacemente il modello concettuale in un ambiente operativo, consentendo la gestione, il monitoraggio e l'esecuzione dei processi aziendali in modo efficiente e affidabile.

3.1 Diagramma di collaborazione BPMN

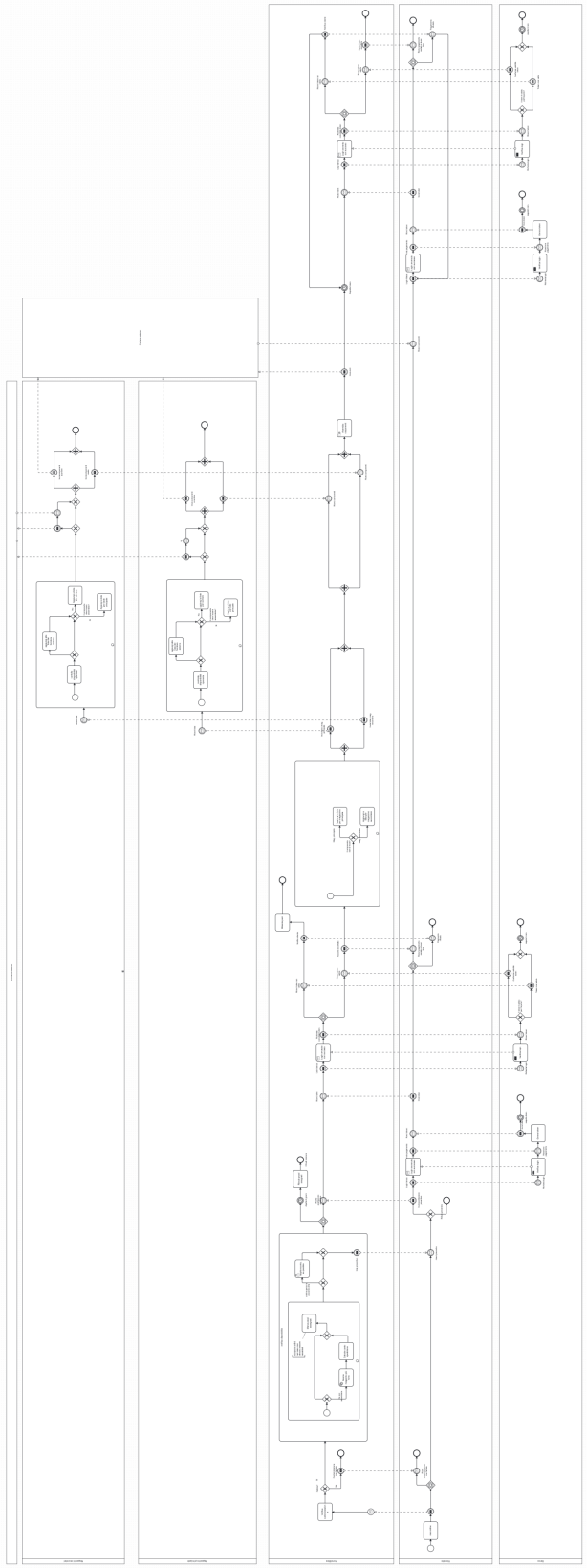

Figura 3.1: Diagramma Completo

3.2 Verifica della disponibilità

Il Sub Task "Verifica Disponibilità" controlla per ogni bicicletta e componente ordinato se sia necessario l'assemblaggio. In caso contrario viene selezionato il componente nel magazzino più vicino alla rivendita in modo da abbassare il costo di spedizione. Successivamente viene calcolato il costo finale ed applicato se possibile, uno sconto¡

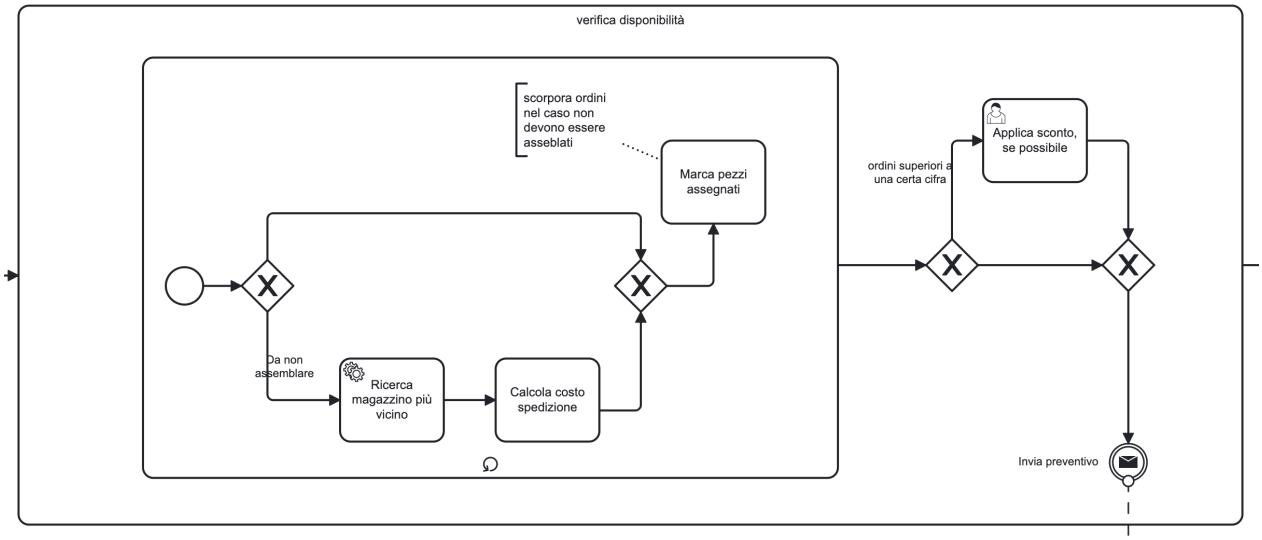

Figura 3.2: Diagramma verifica disponibilità

3.3 Creazione liste per i magazzini

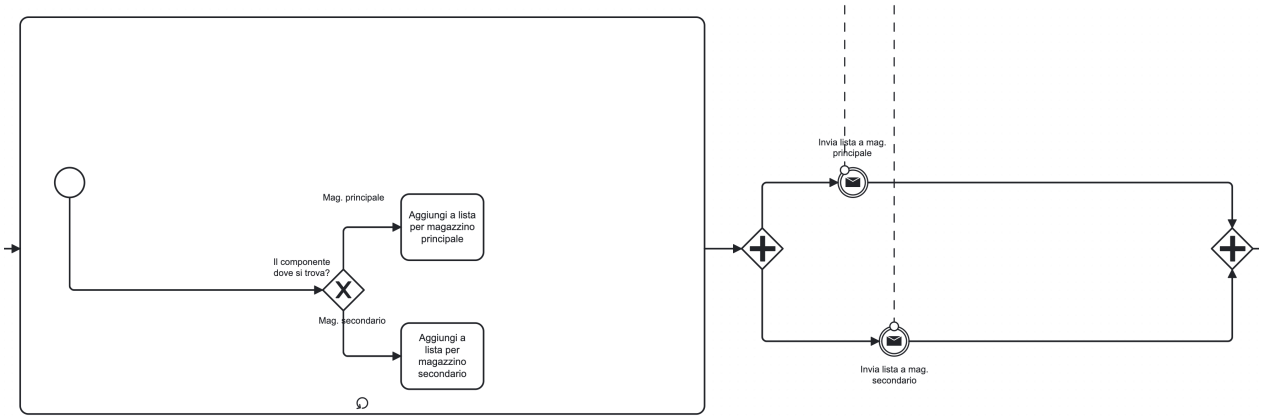

Figura 3.3: creazione liste per i magazzini

3.4 Magazzini

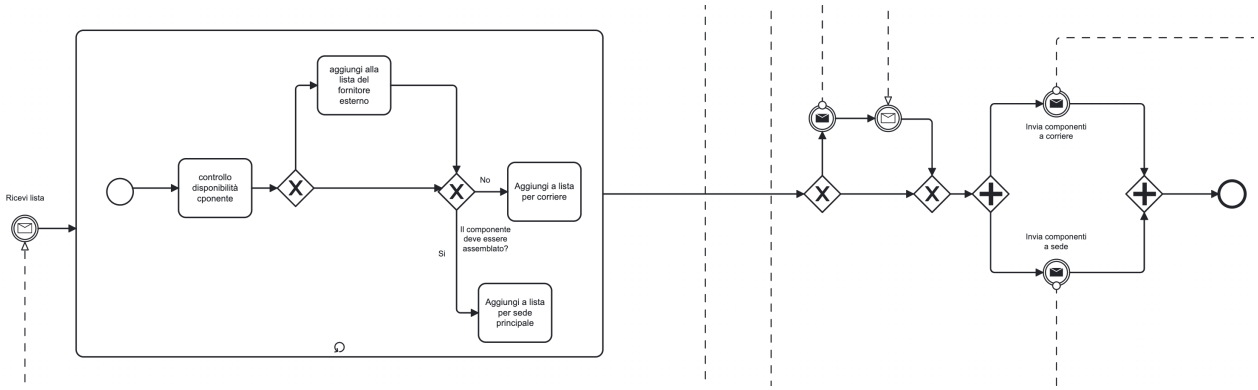

Figura 3.4: Diagramma magazzini

3.5 Pagamento e verifica del token

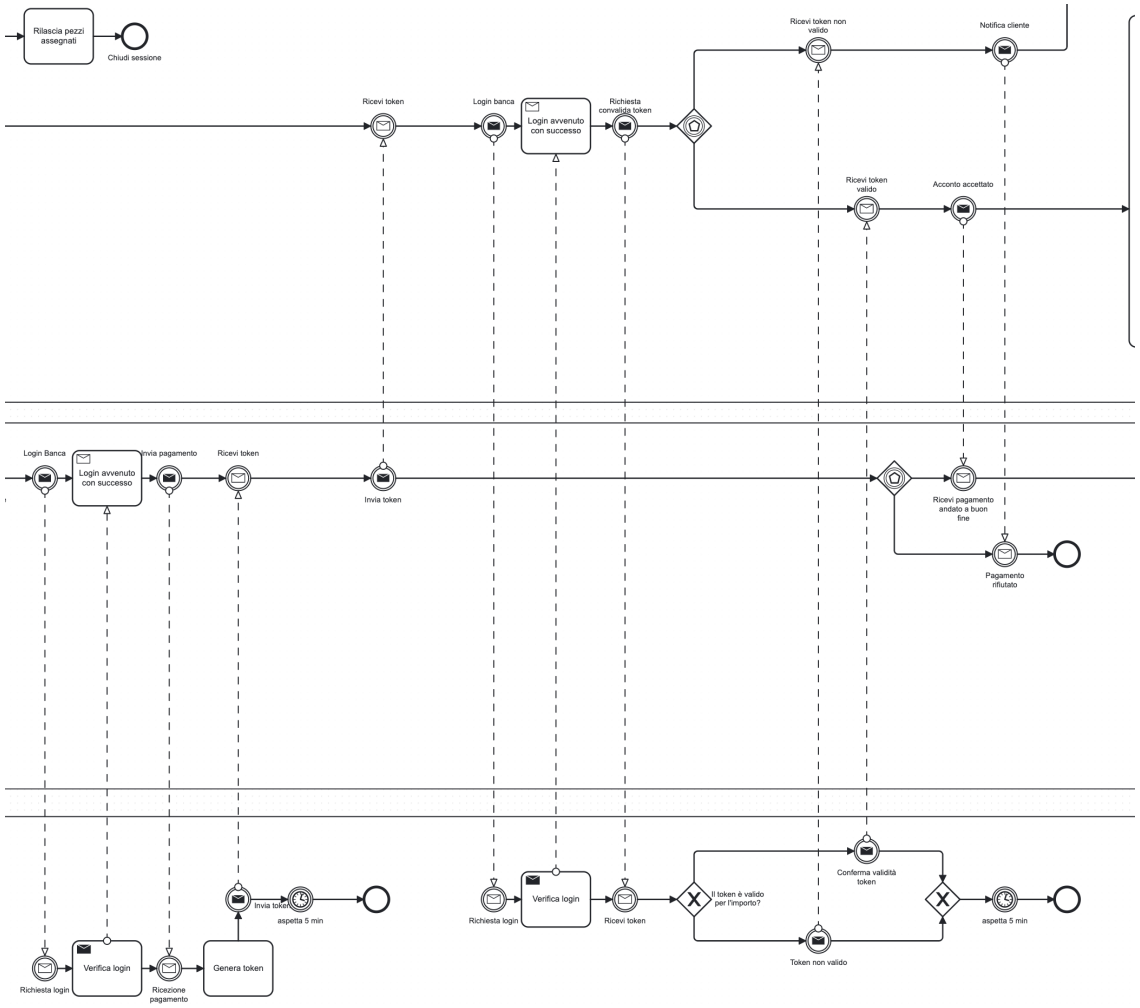

Figura 3.5: Diagramma pagamento alla banca e verifica token

Capitolo 4

Diagramma

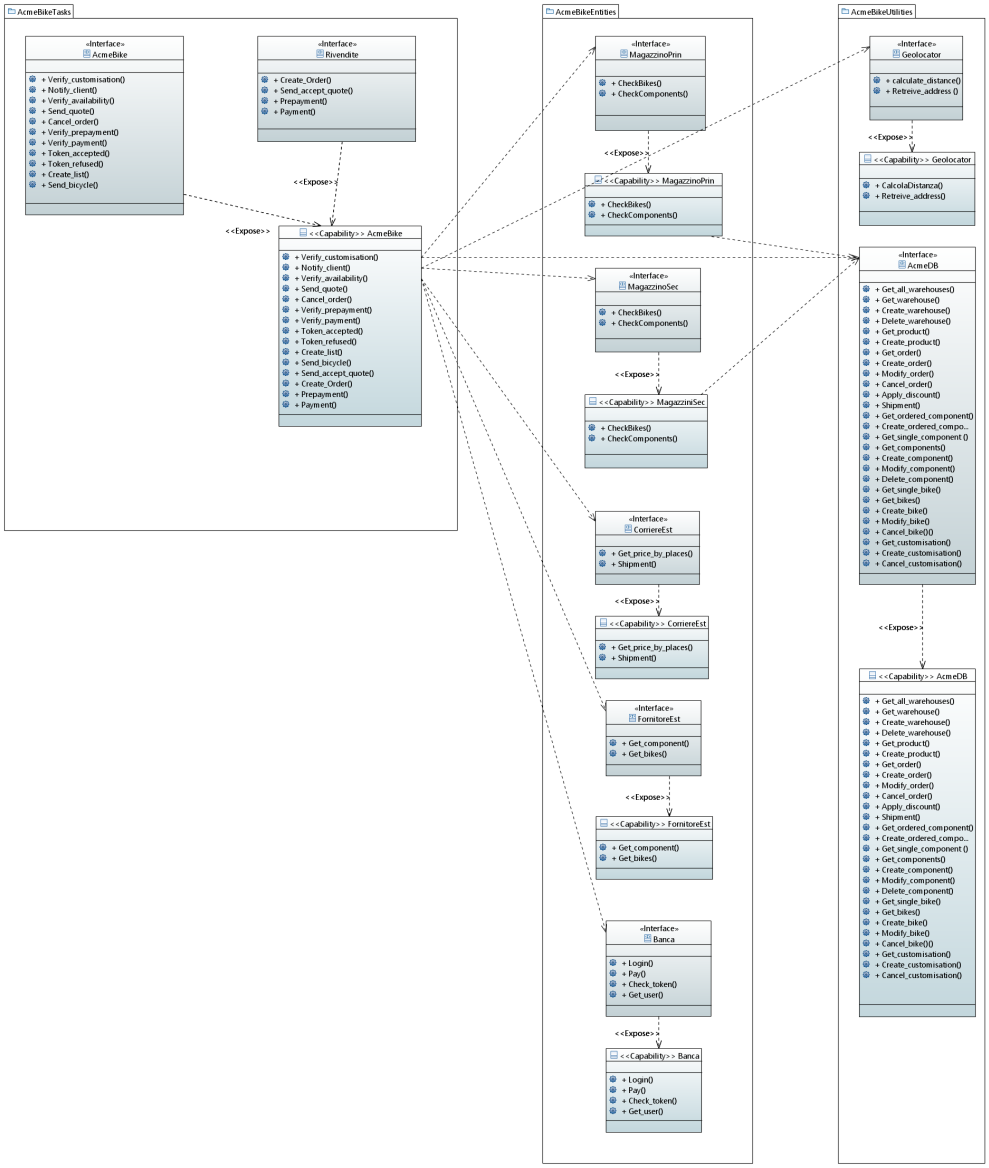

Figura 4.1: Diagramma UML tiny SOA

Capitolo 5

Sviluppo

In questa fase di sviluppo sono stati realizzati i servizi emersi dalla specifica:

- Servizio core AcmeBike che rende accessibili le capabilities realizzate attraverso il BPMS Camunda;
- Servizio di spedizione esterno;
- Servizio di geolocalizzazione;
- Servizio bancario;
- Servizio di fornitura esterno;
- Servizio di gestione dei magazzini Warehouse.

5.1 Diagrammi BPMN esegubili

Come discusso in precedenza nel capitolo 3, il diagramma BPMN generale è utile a scopo documentativo per avere una panoramica su tutto il sistema. Tramite il BPMS Camunda abbiamo realizzano due diagrammi, uno per AcmeBike e uno per le Rivendite, eseguibili per simulare e automatizzare il processo descritto nella traccia.

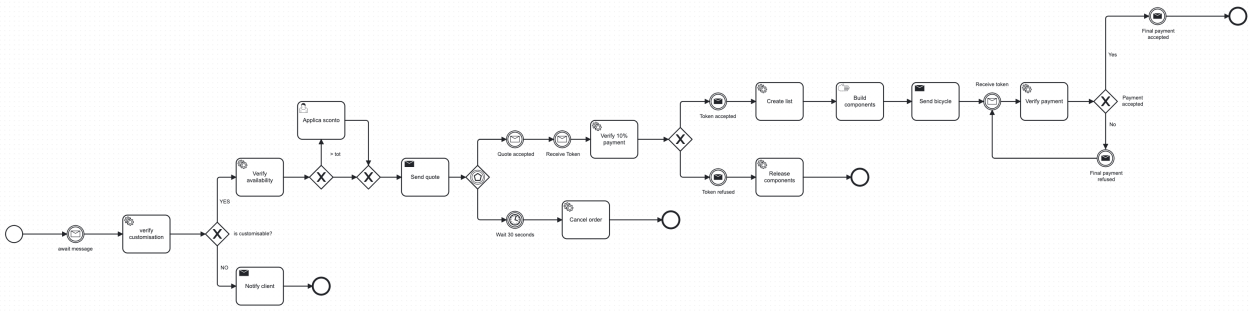

Figura 5.1: Diagramma eseguibile di acmeBikes

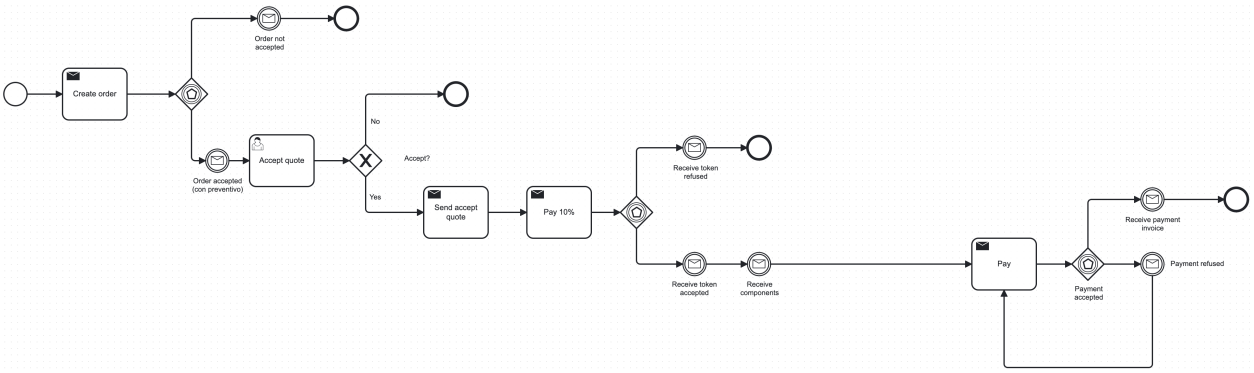

Figura 5.2: Diagramma eseguibile delle rivendite

5.2 Backends

Sono stati implementati i seguenti backend:

- **acmeBikes**: BPMS per la gestione della comunicazione tra AcmeBikes, le rivendite e i vari attori
- **AcmeDB**: backend REST che gestisce l'intera base dati a supporto delle operazioni implementate dai servizi;
- **Courier**: backend REST che simula la ricezione e l'invio di componenti;
- **Geoloc**: backend REST per il calcolo delle distanze geografiche implementato tramite la libreria di python *geopy*. i metodi esposti da tale servizio sono:
  - *calculate_distance* per il calcolo della distanza geografica dati latitudine e longitudine di due punti
  - *retreive_address* che ritorna latitudine e longitudine di un punto dato l'indirizzo
- **Bank**: backend REST per la gestione dei conti bancari, utilizzato per simulare i pagamenti da parte delle rivendite e le autenticazioni dei token.
- **Supplier**: backend REST che simula il fornitore esterno nella ricezione di una lista di ordini e nell'invio dei componenti ai vari magazzini;
- **Warehouse**: backend JOLIE implementato per simulare il lavoro dei magazzini, ripartito in: ricezione della lista dei componenti, richiesta e ricezione di componenti al fornitore esterno se necessario, invio componenti o al corriere esterno o alla sede principale di AcmeBikes.

5.2.1 Worker

La comunicazione tra il BPMS e il codice python viene gestito tramite la libreria **pycamunda**. È stato utilizzato un **worker** che rimane in ascolto per i task da eseguire tramite **camunda**. Più nello specifico il worker supporta alcune funzioni tra cui:

- `subscribe` per iscrivere un topic al worker
- `unsubscribe` per rimuover un topic dal worker
- `run` per eseguire il worker
- `work` funzione specifica per l'esecuzione in parallelo dei task con altri thread

5.2.2 Interface

I service task utilizzati all'interno del BPMS sono stati definiti nel file `interface.py` utilizzando la funzione `subscribe` del worker:

```
    worker.subscribe(
        topic='external_task',
        func=function,
        variables=["process_instance_id", "process_dict"]
    )
```

Di seguito viene fornita l'implementazione della funzione `notify_client.py` del topic, responsabile dell'annullamento dell'ordine:

```
    def notify_client(process_instance_id, process_dict):
    load_dotenv()
    CAMUNDA_URL = os.getenv("CAMUNDA_URL")
    print(f"notify_client {process_instance_id}")
    try:
        # list out keys and values separately
        key_list = list(process_dict.keys())
        val_list = list(process_dict.values())
        # print key with val 100
        position = val_list.index(process_instance_id)
        resale_process_instance_id = key_list[position]

        process_dict[process_instance_id] = resale_process_instance_id

        msg = CorrelateSingle(CAMUNDA_URL, message_name="
            order_not_accepted",
                              process_instance_id=
                                  resale_process_instance_id
                              )
        msg()
    except Exception as e:
        print(e)
        pass
    return {"order_canceled": True}
```

5.2.3 Servizi

I servizi REST sono stati implementati tramite il framework per python **FastAPI**. La struttura generale comprende il file *main.py* dove sono racchiuse le chiamate api e *model.py* dove sono definiti i modelli json da utilizzare nelle chiamate **POST**.

AcmeDB

Tutti i dati relativi alla realtà di AcmeBikes come componenti, biciclette, ordini, componenti ordinati oltre ai dati dei vari magazzini vengono immagazzinati all'interno di un server appositamente creato denominato AcmeDB. Il servizio espone tramite API REST tutte le capability per accedere, creare modificare ed eliminare i records all'interno del database **SQLite**. Tale scelta progettuale è stata preferita ad altre al fine di semplificare la comunicazione interna ad AcmeBikes tra le varie tecnologie.

Bank

Il servizio bancario oltre ad includere tutte le funzionalità **REST** già discusse in precedenza include anche un sistema di autorizzazione per l'accesso alle risorse tramite **JWT**. La creazione e la verifica del JWT di accesso viene fatto tramite le funzioni:
`signJWT`:

```
def signJWT(user_id: str) -> Dict[str, str]:
    payload = {
        "user_id": user_id,
        "expires": time.time() + 600
    }
    token = jwt.encode(payload, JWT_SECRET, algorithm=JWT_ALGORITHM)

    return token_response(token)
```

`verify_jwt`:

```
def verify_jwt(self, jwtoken: str) -> bool:
    isTokenValid: bool = False

    try:
        payload = decodeJWT(jwtoken)
    except:
        payload = None
    if payload:
        isTokenValid = True
    return isTokenValid
```

I dati degli utenti quali username, password, disponibilità e lo storico delle transazioni sono immagazzinati in un database relazionale **SQLite**.

Warehouse

Il sistema di gestione dei magazzini principale e secondari è stato realizzato tramite **Jolie**.

I servizi implementati in **Jolie** sono raggiungibili da AcmeBikes utilizzando il protocollo **SOAP**. Questi servizi sono stati definiti tramite file WSDL generati utilizzando la utility **jolie2wsdl**, che permette di esporre le operazioni dei servizi in modo standardizzato e interoperabile.

All'interno dell'ambiente Jolie, la comunicazione avviene con i servizi di fornitura esterna (supplier), spedizione (courier) e database tramite il protocollo REST.

Per permettere la comunicazione via SOAP tra AcmeBikes e i magazzini è stata definita una **input port** come segue:
`warehouse_input_port`:

```
inputPort MainWarehouseService {
        Location: "socket://localhost:8085"
        Protocol: soap
        Interfaces: WarehouseInterface
}
```

Per le comunicazioni REST a fornitore esterno, corriere esterno e database sono state definite tre **output port** come segue:
`warehouse_output_port`:

```
interface CourierInterface {
    RequestResponse: shipment(ComponentCourierRequest)(string)
}


interface SupplierInterface {
    RequestResponse: supplyComponents(ComponentSupplierRequest)(string)
    RequestResponse: supplyBikes(BikeRequest)(string)
}

interface AcmeDBInterface {
    RequestResponse: component(ModifyComponent)(string)
    RequestResponse: bike(ModifyBike)(string)
}

outputPort CourierService {
    Location: "socket://localhost:8001/"
    Protocol: http {
                .method = "post"
                .format = "json" }
    Interfaces: CourierInterface
}

outputPort SupplierService {
```

```
    Location: "socket://localhost:8003/"
    Protocol: http {
                .method = "post"
                .format = "json" }
    Interfaces: SupplierInterface
}

outputPort AcmeDBService {
    Location: "socket://localhost:8004/"
    Protocol: http {
                .method = "put"
                .format = "json" }
    Interfaces: AcmeDBInterface
}
```

I servizi dei magazzini implementano due metodi: `checkBikes` e `checkComponents`.

`checkBikes`:

```
[checkBikes(bikeRequest)(response){
                i = 0
                for (bike in bikeRequest.bikes) {
                        if (bike.qty < 0){
                                bikeForSupplier.bikes[i] << bike
                                i = i + 1
                        }
                }
                if (i > 0){
                        supplyBikes@SupplierService(bikeForSupplier)(
                           supplierResponse );
                        println@Console( supplierResponse )()
                }
                for (bike in bikeForSupplier.bikes) {
                        request.bike_id = bike.bike_id
                        request.qty = bike.qty * -1
                        bike@AcmeDBService(request)( dbResponse );
                println@Console( dbResponse )()
                }

                response = bikeRequest
        }]{
                println@Console( response )()
        }
```

checkComponents:

```
[checkComponents(componentsRequest)(response){
                i = 0
                for (component in componentsRequest.components) {
                        if (component.qty < 0){
                                componentsForSupplier.components[i] <<
                                    component
                                i = i + 1
                        }
                }
                // Chiedere al fornitore esterno
                if (i > 0){
                        supplyComponents@SupplierService(
                            componentsForSupplier)( supplierResponse );
                        println@Console( supplierResponse )()
                }

                for (component in componentsForSupplier.components) {
                        request.prod_id = component.component_id
                        request.qty = component.qty * -1
                        component@AcmeDBService(request)( dbResponse );
                println@Console( dbResponse )()
                }

                j = 0
                z = 0
                for (component in componentsRequest.components) {
                        if (component.assembleable){
                                componentsForCourier.components[j] <<
                                    component
                                j = j + 1
                        }
                        else {
                                componentsForAcmeBike.components[z] <<
                                    component
                                z = z + 1
                        }
                }
                componentsForCourier.resale_instance_id =
                    componentsRequest.resale_instance_id
                componentsForCourier.contact_resale = false
                // Contattare corriere
                if (j > 0) {
                        shipment@CourierService(componentsForCourier)(
                            courierResponse );
                        println@Console( courierResponse )()
```

```
40                  }
41
42                  response = componentsForAcmeBike
43          }]{
44                  println@Console( response )()
45          }
```

Capitolo 6

Conclusioni

Le conclusioni del nostro progetto riflettono gli obiettivi primari che abbiamo affrontato nel corso della nostra ricerca e sviluppo nell'ambito dell'Ingegneria del Software Orientata ai Servizi. Abbiamo dedicato sforzi significativi per acquisire una solida comprensione dei workflow, delle best practices e delle tecnologie pertinenti, come dimostrato dall'analisi delle fasi di modellazione e sviluppo del nostro progetto. Durante il processo, abbiamo riconosciuto l'importanza di adattare la nostra documentazione e il nostro lavoro di sviluppo per rispondere alle esigenze mutevoli dell'applicazione e dei singoli servizi. Questa flessibilità ci ha permesso di garantire la coerenza e la correttezza del sistema, oltre a fornire una guida per eventuali modifiche future.

In una fase preliminare si sono fatte delle assunzioni e semplificazioni per garantire uno sviluppo a livello simulativo del sistema. Nella prima fase progettuale è stata data un'importanza significativa alle comunicazioni tramite lo sviluppo della coreografia che vede impegnati i vari attori nella comunicazione per l'esecuzione corretta che va dall'inoltro dell'ordine da parte delle rivendite fino all'arrivo della merce a destinazione. La coreografia è stata in seguito proiettata in un sistema di ruoli per evidenziare la correttezza nelle comunicazioni raffinando volta per volta fino ad arrivare al diagramma finale di coreografia BPMN. La coreografia ci ha portati alla seconda fase progettuale nella quale abbiamo modellato l'intero scenario descritto nella traccia, basandoci su quanto emerso dalla coreografia in modo tale da rimanere coerenti ad essa. A questo punto sono state mostrate delle operazioni cardine per la logica di sviluppo del sistema tramite l'analisi di varie parti del diagramma di collaborazione BPMN. La SOA è stata descritta in un diagramma UML tramite i tre profili tinySOA per presentare l'architettura del sistema. L'ultima parte del percorso ha rigurdato lo sviluppo di due diagrammi BPMN, riferiti alle rivendite e ad AcmeBike, eseguibili tramite il BPMS Camunda. Questi digrammi rimangono in ogni caso corenti con il diagrmma BPMN generale creato a scopo documentativo. Dopo aver mostrato i diagrammi per supportare i due processi, siamo passati alla descrizione dei componenti fondamentali del backend utilizzati per comunicare con il BPMS, le interfacce e i servizi che sviluppano la logica.

Siamo fiduciosi che la documentazione prodotta possa servire come solido punto di partenza per futuri miglioramenti e aggiornamenti del sistema AcmeBike. In definitiva, le nostre conclusioni si allineano con gli obiettivi originali del progetto.

Elenco delle figure