

Ingegneria del Software A.A. 2018/2019

Esame 2019-06-17

Esercizio 1 (6 punti)

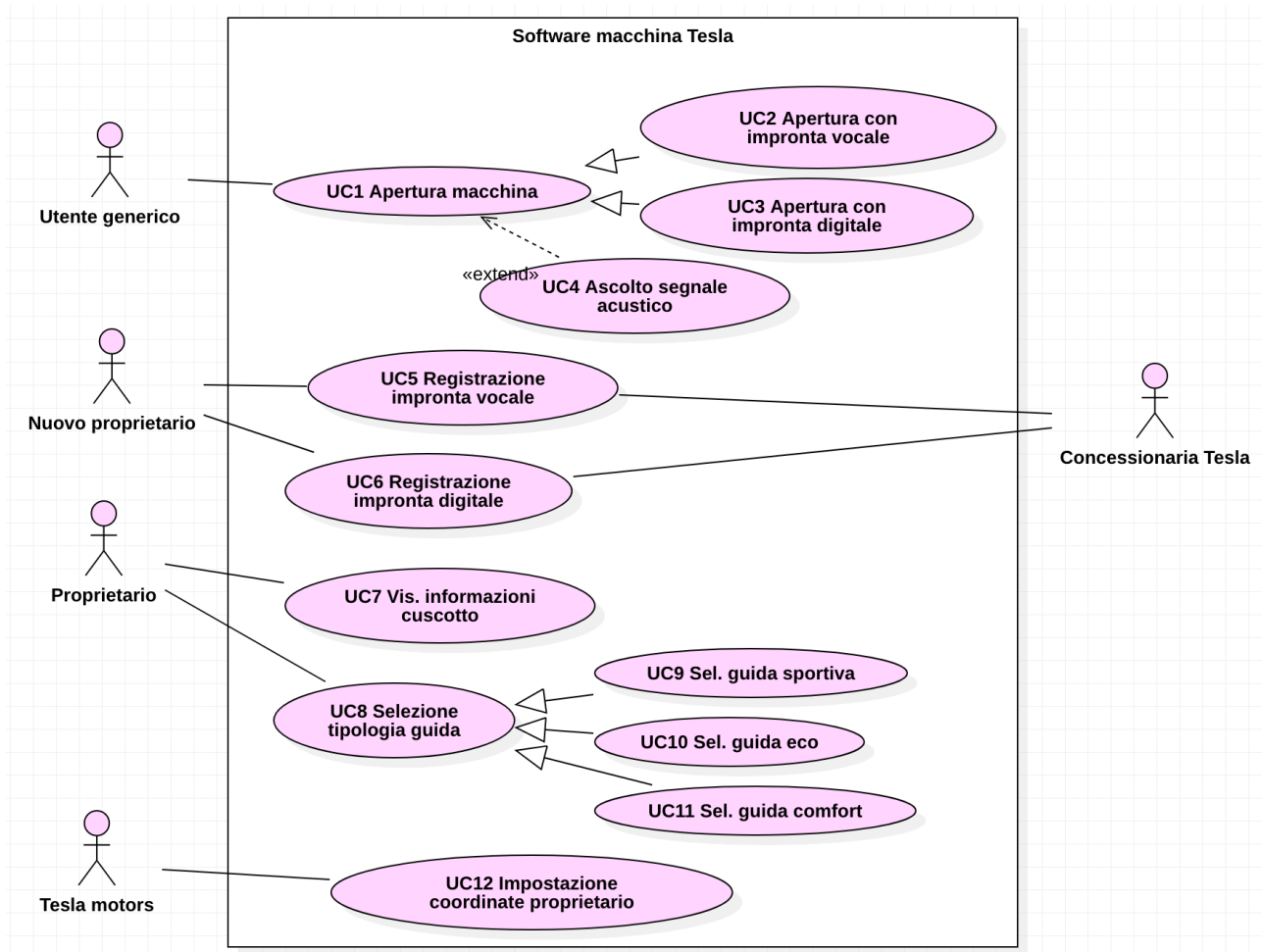
Descrizione

Fin dalla sua nascita nel 2003, Tesla Motors si è caratterizzata come grande innovatore nel campo delle auto elettriche. Il fiore all'occhiello della sua produzione sono le componenti elettroniche e le applicazioni software installate sulle proprie vetture. La tipica auto Tesla può essere aperta dal proprietario utilizzando un'impronta vocale o digitale. Al primo utilizzo in concessionaria, entrambe tali impronte vengono registrate nel software della vettura, in modo da garantirne il futuro riconoscimento. Un successivo tentativo di accesso non riuscito causa l'emissione di un segnale acustico di allarme. Una volta all'interno, il cruscotto della vettura visualizza una serie di informazioni tra le quali: lo stato delle batterie (cariche, da caricare, in carica), la data e l'ora del giorno, e una diagnostica del motore. Per ogni voce della lista delle componenti del motore, il display riporta il nome, e lo stato di usura. Interagendo con l'immagine del motore è possibile selezionare la tipologia di guida desiderata, a scelta tra sportiva, comfort, eco. Utilizzando una *app* dedicata per *smartphone*, è inoltre possibile fornire le proprie coordinate posizionali alla vettura, per farla guidare da sola verso la posizione del proprietario.

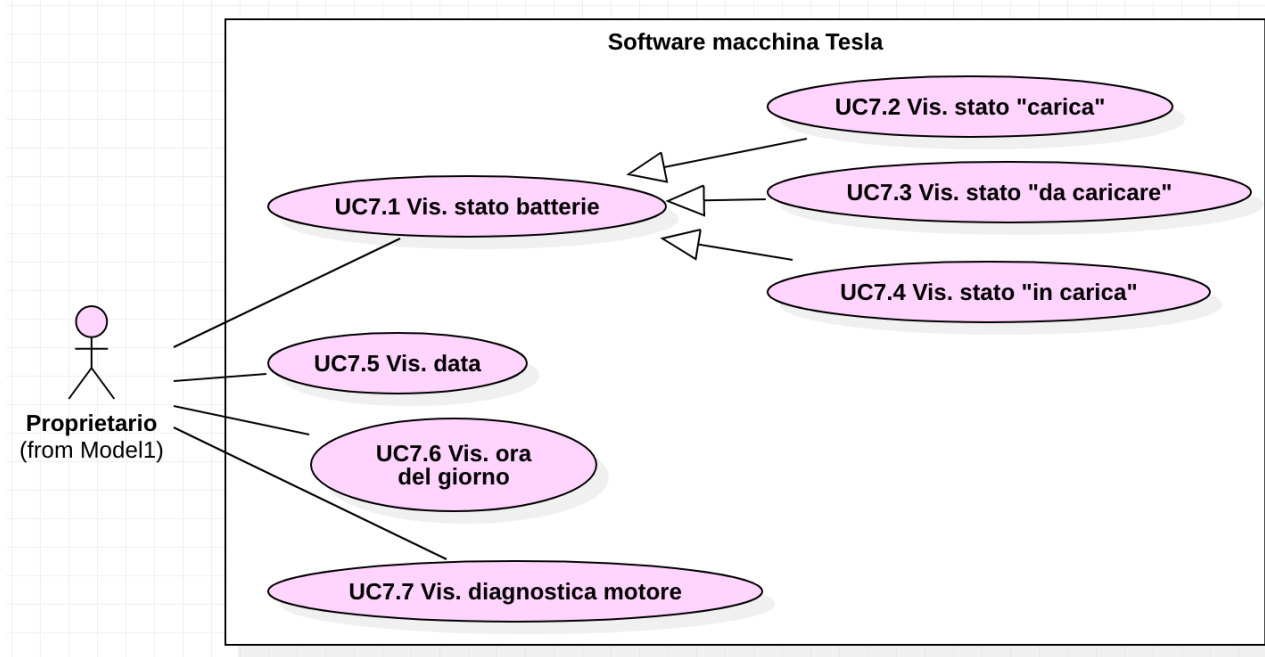
Si utilizzino i diagrammi dei casi d'uso per modellare gli scenari sopra descritti. Non ne è richiesta la descrizione testuale.

Soluzione

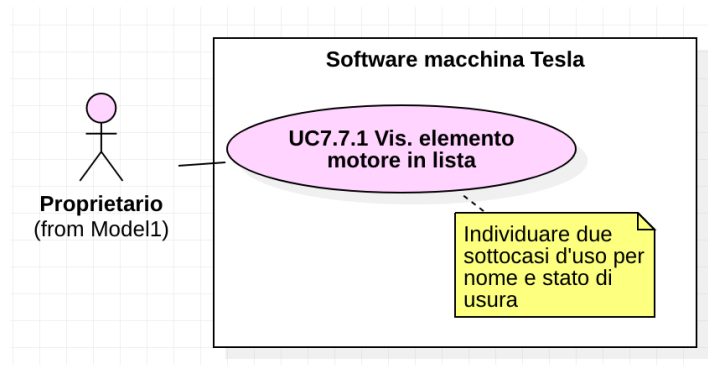
Una possibile soluzione è la seguente. Inizialmente è necessario analizzare il software installato all'interno della macchina. Si noti che si sono individuati tre differenti attori principali relativi al proprietario della macchina. Il sistema auto, infatti, si trova in tre stati completamente differenti durante le interazioni con ognuno di questi attori.



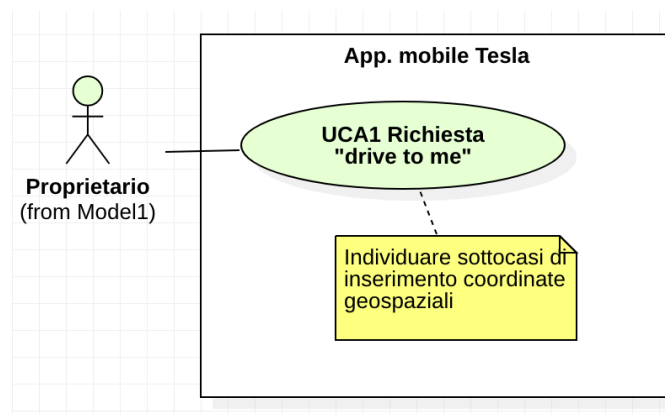
La visualizzazione del cruscotto richiede l'analisi di ulteriori sotto casi d'uso.



E per completare, il diagramma seguente.



L'esercizio richiede anche l'analisi di una funzionalità di un'applicazione mobile esterna alla macchina.



Esercizio 2 (7 punti)

Descrizione

La guida autonoma di Tesla è possibile perché una grande quantità di sensori è installata sulla vettura per fornire al *software* di controllo l'informazione necessaria per conoscere costantemente lo stato dei suoi apparati e dell'ambiente circostante. In particolare, le molte telecamere poste sulla vettura producono una immagine (*frame*) ogni 10 millisecondi. Tale flusso di immagini viene catturato da opportuni *listener*, che successivamente le forniscono a un algoritmo di apprendimento automatico, capace di categorizzare le cose o le persone rilevate intorno alla vettura. Qualora venga rilevato un pedone entro 20 metri dalla traiettoria attuale della vettura, tale informazione viene notificata a una centralina di controllo che attiva immediatamente i sistemi di sicurezza attiva, i quali arrestano la vettura. La centralina è in grado di eseguire diverse operazioni su molteplici componenti dell'auto, pur mantenendo un'unica interfaccia di esecuzione, ciò che garantisce massima estensibilità del *software*.

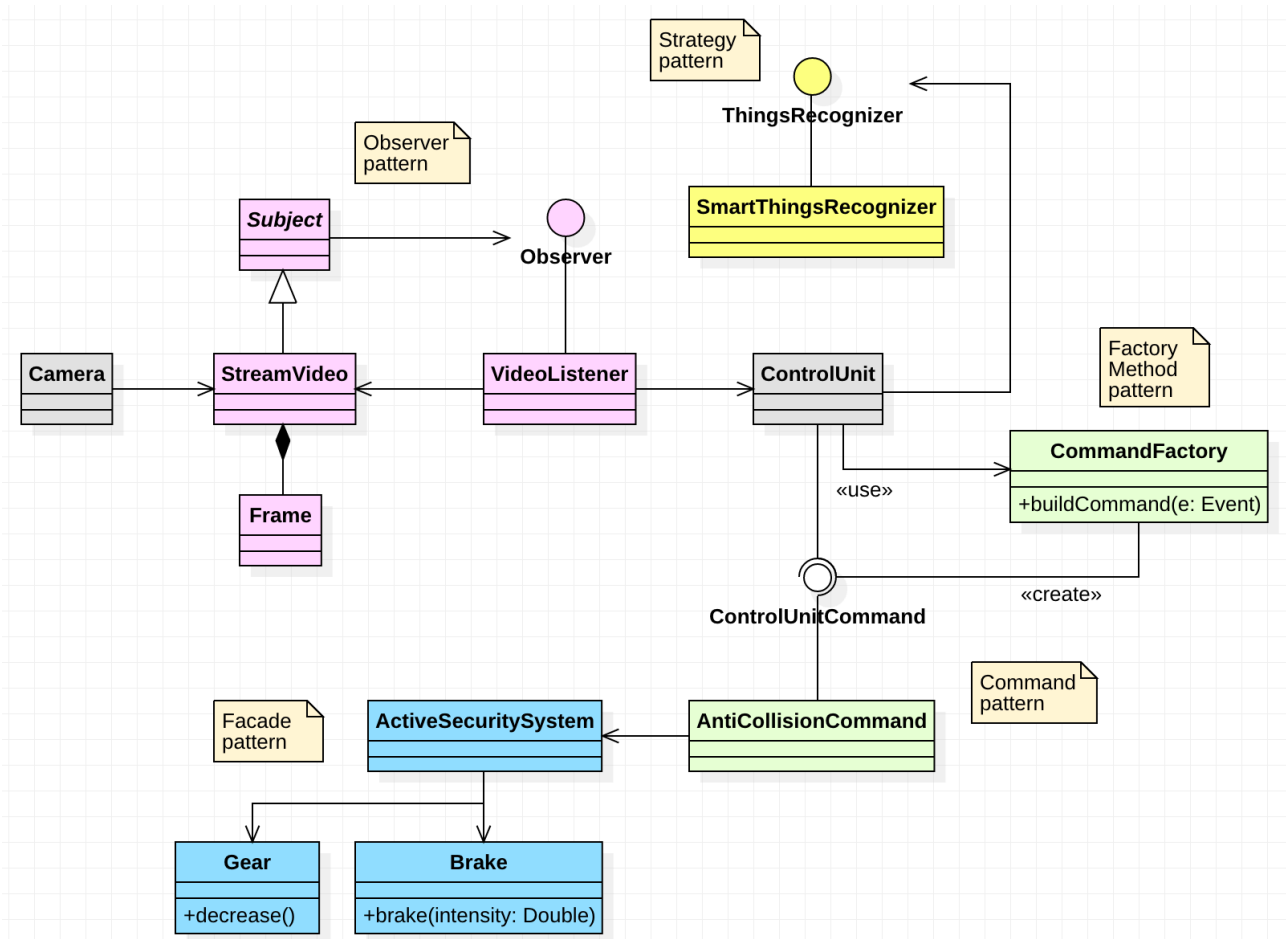
Si modelli il sistema sopra descritto mediante un diagramma delle classi e i *design pattern* a esso pertinenti. Immaginando che un pedone venga rilevato in rotta di collisione con la vettura, si utilizzi un diagramma di sequenza per descrivere l'interazione fra le componenti individuate, fino al completo arresto della vettura.

Soluzione

La soluzione prevede l'utilizzo dei seguenti *design pattern*:

- Observer pattern, per acquisire le informazioni dalle telecamere dall'auto
- Strategy pattern, per applicare l'algoritmo di riconoscimento degli ostacoli

- Command pattern, per gestire in modo standard le operazioni all'interno della centralina di controllo
- Factory method pattern, per costruire i comandi da eseguire sulla base degli eventi scatenati nella centralina
- Facade pattern, per gestire tutte le operazioni necessarie sulle varie componenti della macchina per implementare la sicurezza attiva



Il diagramma di sequenza associato ne deriva in modo lineare.

Esercizio 3 (3 punti)

Descrizione

Poiché il possessore di una vettura Tesla si aspetta elevate prestazioni da tutti gli apparati, meccanici ed elettronici, della vettura, il *software* installato in essa utilizza ampiamente la *reactive programming*. Basata sul concetto di *observer*, questo paradigma di programmazione permette di reagire prontamente a eventi esterni. Per questo motivo, il *reactive programming* è utilizzato proficuamente, per esempio, per gestire i sensori che rilevano la velocità della macchina.

Riferendosi al pattern *Observer*, fornire il diagramma di sequenza di una possibile interazione fra i sensori di velocità e un *listener* che ne utilizza le informazioni.

Soluzione

Per semplicità, si suppongano unicamente messaggi sincroni. I tipi che partecipano al *pattern Observer* sono `Speed Sensor` e `Speed Listener`, rispettivamente in qualità di *subject* e *observer*.

