

# Drone mission

Carlo Antenucci, Leonardo Iannacone, Gonzalo Junquera

6 marzo 2013

# Indice

<b>1</b>	<b>Introduzione</b>	<b>4</b>
<b>2</b>	<b>Visione</b>	<b>5</b>
<b>3</b>	<b>Obiettivi</b>	<b>6</b>
<b>4</b>	<b>Requisiti</b>	<b>7</b>
4.1	Lo scenario applicativo . . . . .	7
4.2	Il lavoro da svolgere . . . . .	8
4.3	Remark . . . . .	8
<b>5</b>	<b>Analisi dei requisiti</b>	<b>10</b>
5.1	Use cases . . . . .	10
5.2	Glossario . . . . .	11
5.3	Scenari . . . . .	12
5.4	(Domain) Model . . . . .	12
5.5	Test plan . . . . .	12
<b>6</b>	<b>Analisi del problema</b>	<b>13</b>
6.1	Logic architecture . . . . .	13
6.2	Abstraction gap . . . . .	13
6.3	Risk analysis . . . . .	13
<b>7</b>	<b>Piano di lavoro</b>	<b>14</b>
<b>8</b>	<b>Progetto</b>	<b>15</b>
8.1	Struttura . . . . .	15
8.2	Interazione . . . . .	15
8.3	Behavior . . . . .	15
<b>9</b>	<b>Implementazione</b>	<b>16</b>
<b>10</b>	<b>Testing</b>	<b>17</b>
<b>11</b>	<b>Deployment</b>	<b>18</b>

<i>INDICE</i>	2
<b>12 Maintenance</b>	<b>19</b>

# Elenco degli algoritmi

# Capitolo 1

## Introduzione

## Capitolo 2

# Visione

Per esplorare territori difficilmente accessibili, sia per condizioni ambientali che per disastri naturali, potrebbe essere utile un drone teleguidato da una centrale operativa che rilevi i dati territoriali mediante fotografie e li comunichi alla stessa.

## Capitolo 3

# Obiettivi

L'applicazione intende fornire alla protezione civile un maggiore supporto per l'esplorazione territoriale senza la necessità di mettere a repentaglio vite umane.

## Capitolo 4

# Requisiti

### 4.1 Lo scenario applicativo

La protezione civile decide di inviare su un luogo difficilmente accessibile un aeromobile senza pilota (*drone*), capace di operare in modo teleguidato. Il drone è dotato di un insieme di *sensori di stato* in grado di rilevare la velocità corrente (*speed*) e il carburante disponibile (*fuel*). Il drone dispone anche di un dispositivo *GPS* in grado di determinarne la posizione in termini di latitudine e longitudine.

Il compito del drone è scattare fotografie del territorio ogni DTF ( $DTF > 0$ ) secondi e inviare le immagini a un server installato presso una unità operativa. Il server provvede a memorizzare le immagini ricevute (in un file o in un database) associandole ai dati dei sensori di stato disponibili al momento dello scatto della foto. Il server provvede inoltre a visualizzare su un display dell'unità operativa i valori di stato ricevuti dal drone in una dashboard detta *DroneControlDashboard*.

La *DroneControlDashboard* viene concepita come un dispositivo composto di due parti: una parte detta *GaugeDisplay* e una parte detta *CmdDisplay*. La parte *GaugeDisplay* della *DroneControlDashboard* visualizza i dati provenienti dai sensori del drone riconducendoli ciascuno a uno specifico strumento di misura; uno *Speedometer* (velocità in km/h) un *Odometer* (numero di km percorsi) un *FuelOmeter* (livello corrente di carburante in litri) e un *LocTracker* (posizione del drone). La *GaugeDisplay* può visualizzare i dati in forma digitale e/o grafica; la posizione viene preferibilmente visualizzata fornendo una rappresentazione del drone su una mappa del territorio. La parte *CmdDisplay* della *DroneControlDashboard* include pulsanti di comando per fissare la velocità di crociera (*setSpeed*) avviare (*start*) e fermare (*stop*) il drone<sup>1</sup> e per incrementarne (*incSpeed*) e decrementarne (*decSpeed*) la velocità corrente di una quantità prefissata DS ( $DS > 0$  km/h).

---

<sup>1</sup>Il drone si suppone abbia un sistema di controllo capace di eseguire i comandi di *start* e di *stop* in modo opportuno.



I dati dei sensori del drone sono anche resi disponibili sugli smart device in dotazione al responsabile della protezione civile (*Chief*) e al comandante (*Commander*) della unità operativa. Ogni smartdevice provvederà a visualizzare (su richiesta dell'utente) i dati in una dashboard (*SmartDeviceDashboard*) opportunamente definita per lo specifico dispositivo, preferibilmente in modo analogo alla *GaugeDisplay*.

Il server deve operare in modo che :

- la missione del drone possa iniziare solo dopo che il drone ha dato conferma della ricezione del comando *setSpeed* che fissa la velocità iniziale di crociera;
- la speed del drone sia sempre compresa tra due valori-limite prestabiliti *speedMin* a *speedMax*<sup>2</sup>;
- all'avvio di ogni missione, ogni smartdevice **Android** sia messo in grado di generare una *notification* all'utente, la cui selezione provvede ad aprire una applicazione che mostri la *SmartDeviceDashboard*.
- gli smartdevice siano in grado di visualizzare lo stato del drone anche in caso di guasto del server centrale.
- il comando di stop sia inviato in modo automatico non appena il livello del carburante risulta inferiore a un livello prefissato *MinFuel*.

## 4.2 Il lavoro da svolgere

In questo quadro, si chiede di definire il software da installare sul server della unità operativa e su smartdevice dotati di sistema operativo Android<sup>3</sup>. Opzionalmente: si chiede di definire uno strumento capace di visualizzare le informazioni memorizzate dal server dopo una missione del drone. Si chiede anche di costruire un opportuno simulatore delle attività del drone con riferimento ai seguenti parametri:

Parametri per la simulazione del drone

DTF=5 sec, DS=10 km/h, livello fuel iniziale = 30 litri  
 livello minimo fuel per operatività: MinFuel = 0,5 litri  
 speed di crociera compresa tra: speedMin=60 e speedMax=120 km/h  
 consumo di carburante = (speed \* 30) litri/h  
 percorso del drone: in linea retta a una quota fissa di 100m.

## 4.3 Remark

Si ricorda che l'obiettivo del lavoro non è solo la produzione di un sistema software in grado di soddisfare i requisiti funzionali ma anche (e in primis)

<sup>2</sup>Le fasi di decollo e atterraggio sono qui ignorate.

<sup>3</sup>Per il primo protoipo lo smartdevice può essere un computer convenzionale.

il rapporto tra il prodotto e il processo adottato per generarlo. 2 Le fasi di decollo e atterraggio sono qui ignorate. 3 Per il primo protoipo lo smartdevice può essere un computer convenzionale

## Capitolo 5

# Analisi dei requisiti

### 5.1 Use cases

Dalle specifiche del committente si è capito che questi desidera il sistema fornisca tre funzionalità principali: ricezione di informazioni territoriali, controllo della missione e ricezione di informazioni relative ai sensori di stato del drone.

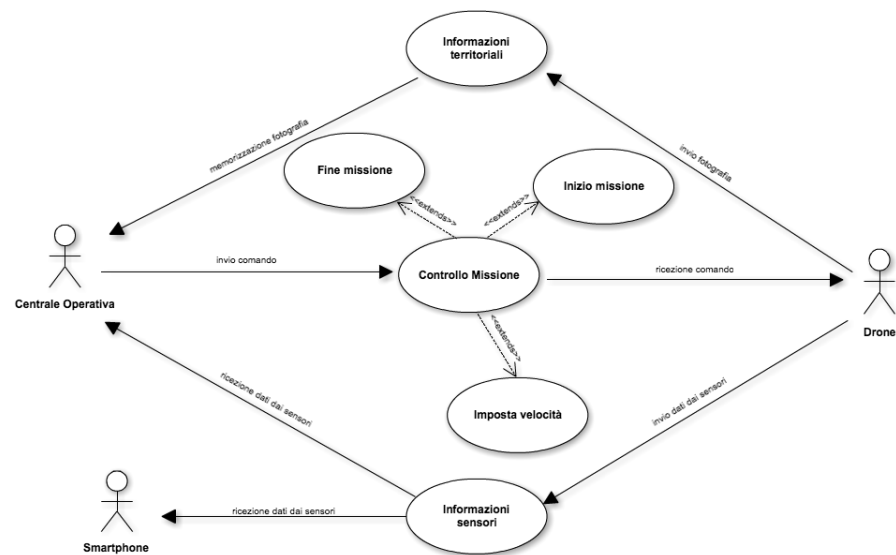


Figura 5.1: Use case

## 5.2 Glossario

TERMINE	SIGNIFICATO
<i>Centrale Operativa</i>	Elemento esterno al sistema da sviluppare. Ha il compito di controllare la missione, ricevere dal drone informazioni relative ai suoi sensori e memorizzare le fotografie scattate.
<i>Smartphone</i>	Elemento esterno al sistema. Consente al Chief e al Commander di avere informazioni sullo stato del drone.
<i>Drone</i>	Elemento esterno al sistema. È un velivolo privo di pilota che ha il compito di esplorare un territorio difficilmente accessibile, di comunicare i dati relativi ai suoi sensori e di inviare ad intervalli di tempo regolari fotografie dell'ambiente esplorato.
<i>Fotografie territoriali</i>	Immagine jpg, acquisite dal drone tramite una fotocamera, che riposta informazioni relative alle condizioni ambientali del luogo esplorato e che la centrale operativa provvederà a memorizzare.
<i>Sensori</i>	Elementi attivi del sistema. Inviando alla centrale operativa informazioni sullo stato del drone, quali chilometri percorsi, velocità attuale, quantità di carburante residuo e le coordinate geografiche del punto in cui si trova.
<i>DroneControlDashboard</i>	Elemento del sistema che consente alla centrale operativa di visualizzare le informazioni ricevute dal drone (GaugeDisplay) e, allo stesso tempo, di inviare al velivolo comandi (CmdDisplay).
<i>GaugeDisplay</i>	Componente della DroneControlDashboard che consente la visualizzazione delle informazioni del drone. È composta da una mappa su cui viene visualizzata la sua posizione e da tre strumenti di misura che riportano i dati rilevati dei sensori, sia in forma analogica che digitale.
<i>Odometer</i>	Strumento di misura che consente la visualizzazione del numero di chilometri percorsi dal drone.
<i>Speedometer</i>	Strumento di misura che consente la visualizzazione della velocità attuale del drone.
<i>FuelOmeter</i>	Strumento di misura che consente la visualizzazione della quantità di carburante disponibile nei serbatoi del drone.

<i>CmdDisplay</i>	<p>Elemento del sistema che consente alla centrale operativa di inviare comandi al drone attraverso una serie di pulsanti. I comandi consentono di:</p> <ul style="list-style-type: none"><li>– Iniziare la missione</li><li>– Terminare la missione</li><li>– Impostare velocità di crociera</li><li>– Aumentare velocità</li></ul>
-------------------	--

### 5.3 Scenari

### 5.4 (Domain) Model

### 5.5 Test plan

## Capitolo 6

# Analisi del problema

6.1 Logic architecture

6.2 Abstraction gap

6.3 Risk analysis

## Capitolo 7

### Piano di lavoro

## Capitolo 8

# Progetto

8.1 Struttura

8.2 Interazione

8.3 Behavior



## Capitolo 9

# Implementazione

## Capitolo 10

# Testing

## Capitolo 11

# Deployment

## Capitolo 12

# Maintenance