

POLITECNICO DI MILANO
Scuola di Ingegneria Industriale e dell'Informazione
Corso di Laurea Magistrale in Ingegneria Informatica



A service-oriented architecture for Unmanned Aerial Vehicles

Relatore: Prof. Luca Mottola

**Tesi di Laurea di:
Mattia Moretta, matricola 771007**

Anno Accademico 2012-2013

*Ai miei genitori,
senza i quali non avrei mai potuto raggiungere quest'importante traguardo.*

Abstract

Il mercato delle tecnologie disponibili ad usi civili è in continua espansione e sempre più le grandi aziende investono nel riprogettare dispositivi già esistenti dedicati ad utilizzi specifici, in particolar modo militari, in apparecchi acquistabili da utenti comuni. Una delle categorie maggiormente interessata a continui sviluppi è quella relativa agli Aereomobili a Pilotaggio Remoto (APR), dispositivi che grazie alla loro versatilità riescono ad offrire un abile rimpiazzo a velivoli con a bordo il pilota e fornire servizi difficilmente ottenibili attraverso l'utilizzo di altri strumenti. L'ultimo decennio è inoltre caratterizzato dalla progressiva migrazione di unità funzionali, presenti soprattutto in dispositivi di utilizzo più comune da parte degli utenti quali laptop, smartphone e tablet, al mondo Web che tutt'oggi chiamiamo *Cloud Computing*. Esso identifica un insieme di tecnologie che permettono, tipicamente sotto forma di un servizio offerto da un provider al cliente, di memorizzare, archiviare ed elaborare dati grazie all'utilizzo di risorse hardware e software distribuite virtualizzate in rete in un'architettura tipica Client-Server.

La mia tesi consiste nella definizione di un'architettura basata sull'utilizzo di tecnologie a servizi che consenta un'efficace interazione dell'utente con i principali impieghi che un APR possa offrire, definiti contestualmente dopo un'accurata ricerca e consultazione di casi d'uso realizzati fino ai giorni nostri. I principali vantaggi che da essa derivano sono la riusabilità del codice definito, l'indipendenza dalla piattaforma di esecuzione dei servizi e la sua facile integrazione all'interno di ambienti complessi quali Business Process e sistemi di Workflow Management. Quest'ultima importante peculiarità viene dimostrata attraverso la realizzazione di un caso d'uso reale adottando un'implementazione di Mashup come interfaccia utente, concetto nato dal paradigma Software a Service nel quale dati provenienti da diverse fonti remote interagiscono tra di loro per raggiungere uno scopo comune, dando vita ad un applicativo Web che diventa a sua volta un nuovo servizio.

Ringraziamenti

Il ringraziamento più grande è indirizzato senza dubbio al professor Luca Mottola per il suo prezioso ed indispensabile aiuto a supporto della realizzazione dell'intera opera di tesi. E' doveroso inoltre menzionare fin da subito il collega Luca Chieppa per avermi fornito il tool necessario all'utilizzo dei servizi Web ed avermi aiutato durante la stesura dell'elaborato, oltre a Jessica e Fausto per l'ottima realizzazione degli esperimenti ottenuta insieme e per la pazienza nei miei confronti durante gli intoppi tecnici affrontati.

I ringraziamenti proseguono poi a tutte quelle persone che, inconsciamente o meno, mi hanno supportato e fatto crescere durante questi ultimi anni di studi. Partendo dai coinquilini con i quali ho condiviso gran parte della vita Milanese quali Libu e Macca passando alle persone che mi sono state più accanto nella vita universitaria quali Memmo e Jack, proseguendo alle colleghe di lavoro degli uffici al Politecnico di Milano quali Megghy, Barbara ed Elena arrivando infine a tutti gli amici ovvero Perri, Balla, Roby, Misc, Corvice, Monce, Aisack, Matta, Dade, Elia, Timothy ecc.. con i quali ho condiviso momenti speciali durante i weekend in Valtellina e negli svariati viaggi all'estero ai quali ho partecipato.

Elenco delle figure

2.1	Vari modelli di velivoli pilotati da remoto	7
2.2	Visione aerea di superfici stradali	10
2.3	Esempio di architettura di scambio dati tra APR e stazioni base	11
2.4	L'elicottero Camcopter	11
2.5	Classico sistema di campionamento dati metereologici	13
2.6	Esempio di mini APR	13
2.7	Mappa delle zone di coltivazione suddivise per grado di matu- rezza del prodotto	14
2.8	Analisi digitale del ghiacciaio effettuate a distanza di un anno	15
2.9	Il velivolo sviluppato da Barnard Microsystem	17
2.10	Videocamere visuali ed infrarosse montate sul velivolo Heliv .	18
2.11	Risultato di un'immagine segmentata	19
3.1	Incendio su ortofoto in falsi colori (in verde perimetro fotoin- terpretato, in giallo perimetro GPS) e relativa mappa.	22
3.2	Elementi di interesse presenti in una mappa utilizzata da ar- cheologi	23
4.1	Architettura a due livelli di deployment	31
4.2	Architettura a due livelli di deployment con elaborazione del servizio eseguita nel dispositivo di controllo	32
4.3	Architettura a due livelli di deployment con elaborazione del servizio eseguita nel velivolo	33
4.4	Architettura a due livelli di deployment con elaborazione dati eseguita nel dispositivo di controllo	34
4.5	Architettura a due livelli di deployment con una semplice el- aborazione dati eseguita nel velivolo	35
4.6	Architettura a tre livelli di deployment con elaborazioni servi- zio e dati eseguite da una macchina server	36

5.1	Schema dei componenti del Parrot AR Drone 2.0	39
5.2	Esempio di utilizzo della classe Control Center presente nella libreria YaDrone	41
5.3	Esempio di utilizzo del servizio di modellazione area	46
5.4	Esempio del servizio di modellazione area con raggruppamento punti pari a 2 unità	47
5.5	Interazioni componenti software del sistema nel caso di utilizzo del servizio Sample	53
5.6	Sequence Diagram relativo all'utilizzo del servizio Sample . . .	53
5.7	Interazioni componenti software del sistema nel caso di utilizzo del servizio Search	55
5.8	Sequence Diagram relativo all'utilizzo del servizio Search . . .	55
5.9	Interazioni componenti software del sistema nel caso di utilizzo del servizio Track	56
5.10	Sequence Diagram relativo all'utilizzo del servizio Track	57
6.1	Esempio relativo a punti di connessione di blocchi definiti in Social Mashlight.	62
6.2	Esempio di flusso di processo valido.	63
6.3	GridMeBundle.	64
6.4	Sticker dedicato alla gestione di una galleria di immagini. . . .	69
6.5	Sticker dedicato al caricamento remoto delle immagini sul portale Web Flickr	69
6.6	Sticker dedicato al controllo dell'APR per le sue funzionalità. .	71
6.7	Sticker dedicato alla visualizzazione dei risultati dei servizi di Sample, Search e Track all'interno di una griglia	72
6.8	Sticker dedicato al controllo del APR per le sue funzionalità nella seconda tab	73
6.9	Selezione di un'estensione di area e parametri di ingresso per il servizio di Search.	75
6.10	Punti di interesse visualizzati come Marker all'interno delle mappe di Google Maps.	75
6.11	Sticker dedicato al controllo dell'APR per le sue funzionalità presenti nella terza tab	76
6.12	Selezione delle coordinate sulla griglia per poter indicare i punti nei quali il velivolo dovrà campionare le immagini. . . .	79
6.13	Visualizzazione delle immagini ottenute dal campionamento. . .	80

6.14	Definizione di tutti i parametri per avviare il servizio di Search.	82
6.15	Visualizzazione delle immagini ottenute dall'invocazione del servizio Search.	83
6.16	Definizione dell'ampiezza dell'area di esplorazione, all'interno di una mappa di Google Maps, corrispondente al parametro Area definito per il servizio di Search	84
6.17	Punti interessati dai riscontri ottenuti dall'invocazione del servizio di Search, visualizzati all'interno della griglia dei punti	85
6.19	Caricamento in remoto delle rilevazioni ottenute attraverso l'utilizzo del servizio Search nel portale di condivisione immagini Flickr.	85
6.18	Visualizzazione approfondita dei risultati ottenuti dall'utilizzo del servizio di Search attraverso i marker presenti nella mappa di Google Maps rapportati all'elemento selezionato all'interno della griglia.	86

Elenco delle tabelle

2.1	Categorie di APR suddivise per fasce di peso	7
2.2	Categorie di APR suddivise per fasce di autonomia	8
2.3	Categorie di APR suddivise per fasce di massima altitudine raggiungibile	8
4.1	Tabella con impieghi e relativi input/output delle interfacce proposte	28
5.1	Tabella dei parametri, suddivisi per tipo di dato, definiti per l'utilizzo del servizio Sample	48
5.2	Tabella dei parametri, suddivisi per tipo di dato, definiti per l'utilizzo del servizio Search	49
5.3	Tabella dei parametri, suddivisi per tipo di dato, definiti per l'utilizzo del servizio Track	50
6.1	Campionamenti delle prestazioni del sistema relativi alle ese- cuzioni degli esperimenti del servizio Sample	81
6.2	Campionamenti delle prestazioni del sistema relativi alle ese- cuzioni degli esperimenti riguardanti il servizio Search	87

Listings

5.1	Esempio di un utilizzo minimale del server REST IceBreakerRestServer	43
5.2	Esempio della restituzione del punto (2.0;0.0)	44
5.3	Esempio della restituzione del campione nel punto (2.0;0.0)	44
6.1	Esempio di file config.xml	67

Indice

Abstract	i
Ringraziamenti	iii
1 Introduzione	1
2 Stato dell'arte	5
2.1 Aereomobili a pilotaggio remoto (APR)	5
2.2 Caratteristiche	7
2.3 Impieghi	9
2.3.1 Applicazioni civili generiche	9
2.3.2 Natura ed agricoltura	14
2.3.3 Sicurezza	16
3 Tassonomia	20
3.1 Campionamento	20
3.2 Monitoraggio	21
3.3 Ispezione	21
3.4 Ricerca	23
3.5 Inseguimento	24
4 Architettura	25
4.1 Interfacce a servizi	25
4.2 Due livelli di deployment	31
4.2.1 Invocazione servizio	32
4.2.2 Elaborazione servizio	32
4.2.3 Volo	33
4.2.4 Elaborazione dati	34
4.3 Tre livelli di deployment	36

4.3.1	Elaborazione dei servizi	37
4.3.2	Elaborazione dei dati	37
5	Implementazione REST	38
5.1	A.R. Drone 2.0	38
5.2	YaDrone	40
5.3	IceBreakRestServer	42
5.4	Formato di ritorno dei dati	44
5.5	Servizi esterni	45
5.6	Definizione servizi REST	48
5.7	Design programmatico	51
5.7.1	Sample	52
5.7.2	Search	54
5.7.3	Track	56
6	Caso di studio	58
6.1	Web Meshup: Social Mashlight	59
6.1.1	I componenti fondamentali	61
6.1.2	L'architettura	63
6.2	Integrazione in Social Mashlight	70
6.2.1	Sample	70
6.2.2	Search	73
6.2.3	Track	76
6.3	Esperimenti di utilizzo	78
6.3.1	Scelta architetturale	78
6.3.2	Drone Sample	78
6.3.3	Drone Search	81
6.4	Sommario dei risultati	88
7	Conclusioni e Sviluppi futuri	89
	Bibliografia	91

Capitolo 1

Introduzione

Il mercato delle tecnologie disponibili ad usi civili è in continua espansione e sempre più le grandi aziende investono nel riprogettare dispositivi già esistenti dedicati ad utilizzi specifici, in particolar modo militari, in apparecchi acquistabili da utenti comuni.

Una delle categorie maggiormente interessata a continui sviluppi è quella relativa agli aereomobili a pilotaggio remoto (APR), dispositivi che grazie alla loro versatilità riescono ad offrire un abile rimpiazzo a velivoli con a bordo il pilota e fornire servizi difficilmente ottenibili attraverso l'utilizzo di altri strumenti.

Partendo dallo sviluppo di modelli molto grandi e complessi si è arrivati alla realizzazione di prototipi in scala ridotta, i quali associati a dispositivi di controllo estremamente portatili, si identificano come attrezzature idonee ad un'ampia varietà di impieghi che differiscono per lo più soltanto dalla dotazione di sensori che essi presentano.

L'ultimo decennio è inoltre caratterizzato dalla progressiva migrazione di unità funzionali, presenti soprattutto in dispositivi di utilizzo più comune da parte degli utenti quali notebook, smartphone e tablet, al mondo Web che tutt'oggi chiamiamo '*Cloud Computing*'. Esso identifica un insieme di tecnologie che permettono, tipicamente sotto forma di un servizio offerto da un provider al cliente, di memorizzare, archiviare ed elaborare dati grazie all'utilizzo di risorse hardware e software distribuite virtualizzate in rete in un'architettura tipica Client-Server.

Il salvataggio dei dati è sempre più propenso ad essere indirizzato all'interno di complesse architetture storage presenti nel Web poichè esse garantiscono una miglior affidabilità sulla persistenza dei dati di quanto una singola unità di memoria possa offrire ed un'accessibilità delle informazioni indipendente dalla posizione nella quale l'utente vuole accedervi.

Servizi Web esterni e condivisione delle risorse riescono pertanto ad imporsi come sostituti di software proprietari e complessi apparati hardware, oltretutto difficilmente usufruibili dati gli alti costi di acquisto e gestione, offrendo all'utente una vasta gamma di architetture disponibili da poter utilizzare per raggiungere gli obiettivi definiti all'interno della propria logica dell'applicazione che intende realizzare.

Platform As a Service (PaaS) [16] ovvero distribuzione di piattaforme di elaborazione in remoto e *Software As a Service* (SaaS) [18] ovvero distribuzione di software come servizi Web sono i principali modelli di condivisione che caratterizzano il moderno deployment di applicativi. Essi offrono un elevato livello di astrazione che permette al committente di focalizzarsi sull'obiettivo da raggiungere trascurando notevoli dettagli implementativi, adottando una gestione dinamica del carico e meccanismi di fault-tolerance avanzati che possano garantire costantemente alte prestazioni d'utilizzo.

La mia tesi consiste nella definizione di un'architettura basata sull'utilizzo di tecnologie a servizi, che consenta un'efficace interazione dell'utente con i principali impieghi che un APR possa offrire, definiti contestualmente dopo un'accurata ricerca e consultazione di casi d'uso realizzati fino ai giorni nostri.

I principali vantaggi che da essa derivano sono la riusabilità del codice definito poichè utilizzabile per il controllo di velivoli differenti, l'indipendenza dalla piattaforma di esecuzione dei servizi e la sua facile integrazione all'interno di ambienti complessi. Tali ambienti spaziano dai Business Process [31] per un'integrazione delle funzionalità esposte dal sistema in ambito aziendale, ne è esempio l'utilizzo di APR come corrieri di consegna merce utilizzati dal colosso Amazon [2], fino ad arrivare a sistemi di Workflow Management [26] per una condivisione delle funzionalità esposte attraverso più gruppi di lavoro collaborativi, utile ad esempio in una possibile gestione degli incendi come quella considerata nella terza sezione del secondo capitolo.

A dimostrazione dell'elevato grado di integrabilità del sistema che ci apprestiamo a descrivere presentiamo l'implementazione di un suo caso d'uso reale come esempio di utilizzo, ottenuto attraverso l'utilizzo di una realiz-

zazione di *Mashup* come possibile interfaccia utente: concetto derivato dal paradigma *Software as a service*, sinonimo di applicazione che utilizza dati eterogenei e provenienti da diverse fonti remote i quali interagiscono tra di loro per raggiungere uno scopo comune fornendo un servizio completamente nuovo.

L'integrazione di APR all'interno di un ecosistema più ampio rispetto al semplice applicativo di pilotaggio consente così di integrare tutti i dati da esso forniti all'interno di un flusso di processo molto più vasto.

In termini di utilizzo la piattaforma di visualizzazione considerata consente la creazione di un flusso di informazioni che attraversino più servizi, eventualmente offerti in domini estremamente differenti, attraverso una semplice ed intuitiva interfaccia grafica che mascheri dettagli strettamente legati all'implementazione.

Il velivolo identifica un'ulteriore generica fonte di input/output che definisce un'interfaccia verso tutto ciò che l'APR offre, indipendente dalla tecnologia utilizzata per la sua gestione. Risulta pertanto immediato lo spostamento di tale blocco all'interno di svariate configurazioni di flussi senza dover specificare alcunchè relativo al contesto nel quale si opera: sarà compito di elaborazioni poste a valle occuparsi della gestione dei dati desiderata.

Presentiamo ora l'elenco dei successivi capitoli con una breve descrizione del loro contenuto.

Il secondo capitolo riguarda lo stato dell'arte. Partendo dalla definizione di cosa un APR rappresenta esso prosegue ad analizzare tutte le caratteristiche più importanti che contraddistinguono ogni velivolo. Segue la presentazione dei suoi principali impieghi di utilizzo, suddivisi per ambito di appartenenza, identificati attraverso un'analisi di progetti realizzati da università, istituti di ricerca ed aziende all'avanguardia negli anni intercorsi fino ad oggi.

Il terzo capitolo identifica una possibile tassonomia degli impieghi nell'utilizzo di APR derivata dall'analisi effettuata nel secondo capitolo. Sono state identificate e raggruppate attività, presenti anche in contesti molto differenti, che condividessero la tipologia di parametri che l'utente debba fornire al sistema ed il tipo di risultato da esso ottenuto.

Il quarto capitolo riprende la classificazione degli impieghi ottenuta nel capitolo precedente per procedere con l'identificazione dei singoli input ed output necessari a definire le interfacce il più possibile generiche che debbano essere offerte all'interno del sistema.

Ne presenta poi due possibili architetture di impiego, differenziate per livelli di deployment, elencandone per ciascuna i relativi requisiti funzionali ed evidenziandone i relativi vantaggi e svantaggi.

Il quinto capitolo descrive come l'implementazione dei servizi è stata realizzata. Dopo una presentazione con relativa motivazione delle scelte riguardanti l'utilizzo delle principali tecnologie di supporto utilizzate nel sistema, esso prosegue con il descrivere come si è passati da un livello di astrazione ad interfacce alla definizione e realizzazione del relativo codice nonché sua implementazione a basso livello.

Nell'ultima sezione è poi descritta nel dettaglio come la realizzazione del codice relativo a ciascun servizio è avvenuta, sia per quanto riguarda le strutture logiche di supporto sia riguardo le interazioni logiche presenti.

Il sesto capitolo presenta un caso di studio composto da due esperimenti d'utilizzo, realizzati attraverso l'adozione dell'architettura a due livelli di deployment presentata nel quarto capitolo ed ottenuti con l'effettivo deployment del sistema in un Personal Computer associato all'utilizzo dell'APR indicato nel quinto capitolo; In tal modo, invece di affermare considerazioni basate esclusivamente su dati ricavati da simulazioni, si è potuto confermare il corretto funzionamento del sistema in un ambiente reale, influenzato dall'inaffidabilità del canale di comunicazione wireless e da ulteriori fattori di applicazione quali presenza di vento e limitata autonomia di volo dell'APR.

Introdotta dal software di Mashup utilizzato come supporto all'usabilità dei servizi da parte dell'utente, il capitolo descrive le principali caratteristiche del framework elencandone dapprima le componenti native ed in seguito quelle realizzate appositamente per l'integrazione con il lavoro svolto.

Per ciascun esperimento di utilizzo ne viene poi presentata una descrizione, il flusso di utilizzo relativo alle loro interazioni con l'utente ed alcune considerazioni sulle prestazioni raccolte.

L'elaborato si conclude presentando nell'ultimo capitolo le considerazioni finali ed i possibili sviluppi futuri.

Capitolo 2

Stato dell'arte

Come in ogni elaborato che implichi un'attività di ricerca lo studio presentato nella suddetta tesi avviene esaminando lo stato dell'arte riguardante il contesto nel quale si vuole portare innovazione scientifica. Partendo dalla definizione di cosa un APR rappresenta analizziamo tutte le sue caratteristiche più importanti. Segue la presentazione dei suoi principali impieghi di utilizzo, suddivisi per ambito di appartenenza, identificati attraverso un'analisi di progetti realizzati da università, istituti di ricerca ed aziende all'avanguardia negli anni intercorsi fino ad oggi.

2.1 Aereomobili a pilotaggio remoto (APR)

I termini *Aereomobile a Pilotaggio Remoto* (APR) e *Unmanned Aerial Vehicle* (UAV), si riferiscono a quegli aerei senza pilota a bordo, impiegati prevalentemente per uso militare ma che possono anche avere una funzione civile.

Coniati quasi 100 anni fa, essi traggono le loro origini dal ronzio dei primi modelli somigliante al rumore che fa il maschio dell'ape, il fuco.

Il loro primo tentativo di utilizzo è attribuito all'attacco da parte degli austriaci agli italiani nel 1849. Essi utilizzarono delle mongolfiere cariche di esplosivo le quali furono innalzate e dirette nei pressi di Venezia per poter causare danni sostanziali; fu soltanto un tentativo poiché non essendo ancora realmente comandate da remoto metà delle mongolfiere tornarono nel territorio di partenza colpendo aree sbagliate [9].

Comandato a distanza il primo vero e proprio velivolo fu progettato soltanto nel 1916 con l'obiettivo di scontrarsi con i famosi dirigibili tedeschi Zeppelin. Da questo momento in poi nacque l'idea di poter avere una flotta di dispositivi gestibili da remoto da poter sfruttare con diversi obiettivi militari e strategici.

I primi prototipi furono principalmente impiegati durante la seconda guerra mondiale mentre successivamente, a seguito di svariate perfezionamenti ma soprattutto riduzioni di dimensioni, presero piede anche nell'ambito della guerra fredda.

L'organizzazione mondiale dell'aviazione civile costituita a Chicago nel 1944, ha ultimamente adottato il più corretto termine di Remotely Piloted Aircraft System (RPAS), cioè aerei con sistema di pilotaggio remoto anche se nel gergo comune tutt'oggi li si chiama ancora APR.

Essi sono il componente principale di ciò che viene definito Unmanned Aircraft System (UAS) [23] ovvero sistema di pilotaggio remoto il quale, per poter operare correttamente, necessita di un sistema di controllo, del collegamento di controllo e di eventuale equipaggiamento di supporto.

Sistema di controllo Il sistema di controllo, funzione delle dimensioni del velivolo, può essere una piccola attrezzatura portatile che ne consente il pilotaggio a vista oppure una complessa stazione di terra che può presentarsi come la cabina di pilotaggio di un aereomobile di linea.

Collegamento di controllo Il collegamento di controllo è funzione dell'area che il velivolo può coprire e dei dati che esso deve trasmettere alla stazione ricevente. La tecnologia più avanzata consente addirittura un pilotaggio via satellite per un controllo esaustivo in ogni zona del pianeta.

Equipaggiamento di supporto L'equipaggiamento di supporto infine è strettamente legato all'impiego che l'APR deve compiere.

2.2 Caratteristiche



Figura 2.1: Vari modelli di velivoli pilotati da remoto

Le caratteristiche che identificano ciascuno di essi sono dimensioni, peso, copertura ed autonomia di volo, velocità e carico delle ali; in secondo piano vi sono costo, massima altitudine che essi possono raggiungere e tipo di alimentazione che presentano [22].

Il loro peso può variare molto da modello a modello, partendo da pochi grammi arrivando a svariate tonnellate come illustra la seguente tabella:

Categoria	Peso [Kg]	Esempio
Pesante	200 – 2000	A-160
Medio	50 – 200	Raven
Leggero	5 – 50	RPO Midget
Micro	< 5	Dragon Eye

Tabella 2.1: Categorie di APR suddivise per fasce di peso

Copertura di azione ed autonomia, come illustra la tabella sottostante, sono parametri ovviamente correlati i quali sono fortemente influenzati dall'attività che il velivolo deve compiere e portano a determinare la frequenza di rifornimenti/ricarica necessari:

Categoria	Durata [Ore]	Copertura [Km]	Esempio
Alta	> 24	> 1500	Predator B
Media	5 – 24	100 – 1500	Silver Fox
Bassa	< 5	< 100	Pointer

Tabella 2.2: Categorie di APR suddivise per fasce di autonomia

L'altitudine massima che il velivolo può raggiungere, come illustrato nella tabella che segue, spazia da poche centinaia di metri fino ad arrivare ad una decina di chilometri. Non trasportando passeggeri, i velivoli non sono pressurizzati e possono quindi volare ad altezze precluse agli aerei di linea:

Categoria	Massima Altitudine [m]	Esempio
Alta	> 10000	Pointer
Media	1000 – 10000	Finder
Bassa	< 1000	Darkstar

Tabella 2.3: Categorie di APR suddivise per fasce di massima altitudine raggiungibile

Gli APR di interesse considerati nella nostra tesi, dei quali sono presenti alcuni esempi in figura 2.1 nella pagina precedente, sono i velivoli con peso compreso tra le categorie media e micro, con media-bassa autonomia che sorvolino a basse altitudini la superficie terrestre poichè facilmente acquistabili e dei quali non è necessario ottenere specifiche licenze per la messa in volo.

2.3 Impieghi

Dalla loro storia è facilmente intuibile come i principali impieghi degli APR fossero in gran parte riconducibili ad utilizzi in ambito militare: tali velivoli sono un abile rimpiazzo ai classici aerei durante missioni troppo pericolose [15]; alcuni esempi delle loro applicazioni sono la mappatura di terreni per effettuare simulazioni di attacchi missilistici, il supporto a missioni di ricognizione ed il supporto ad operazioni di trasporto relativo a materiale bellico.

E' tuttavia in continuo aumento il loro impiego in ambito civile, ed è in questa sezione che concentriamo la nostra ricerca. La missione di queste macchine dipende dalle attrezzature, soprattutto sensori di vario tipo e macchine per riprese audio e video ad alta definizione installate a bordo, e dalla capacità della stazione di terra di immagazzinare le informazioni ottenute. Presentiamo di seguito alcuni casi d'uso esaminati in progetti realizzati da università, istituti di ricerca ed aziende all'avanguardia negli anni intercorsi fino ad oggi raggruppandoli per categorie di utilizzo.

2.3.1 Applicazioni civili generiche

Controllo del traffico

Il dipartimento dei trasporti degli Stati Uniti D'America è interessato da anni nell'ottenere statistiche riguardanti il traffico stradale nonché un controllo in tempo reale delle condizioni di viabilità all'interno delle strade americane [28]. Videocamere montate su torri e sensori integrati nel circuito stradale hanno portato ad un sostanziale miglioramento delle condizioni di viaggio, tuttavia il crescente numero delle vetture presenti in circolazione rende queste tecnologie superate. In primo piano entrambe le tipologie di dispositivi offrono un controllo del traffico localizzato soltanto in determinati punti, luoghi dai quali è infattibile mobilitare la struttura di acquisizione dati. Inoltre esse non forniscono alcuna informazione riguardante traiettorie, percorsi di viaggio e quindi ingorghi presenti nella rete stradale.

La visione aerea a corto raggio come nell'esempio in figura 2.2 nella pagina successiva consente di ottenere un elevato livello di dettaglio



Figura 2.2: Visione aerea di superfici stradali

che permette una miglior sorveglianza ed un avanzato controllo degli incidenti. In un primo momento sono stati considerati i satelliti come fonte di acquisizione immagine, ma la presenza di nuvole ed il continuo riposizionamento necessario per mantenere la loro presenza sulla zona da controllare hanno portato a considerare dispositivi che potessero sorvolare la zona interessata da molto più vicino. Sono quindi stati impiegati piccoli velivoli con a bordo un pilota che effettuassero percorsi di ricognizione, ma anch'essi sono risultati limitati poiché difficili condizioni climatiche e situazioni ostili dell'area da sorvolare rendono molto difficile l'acquisizione dei dati. Gli APR possono ovviare a tutte le limitazioni precedentemente descritte, fornendo dati con una velocità maggiore rispetto a quanto si possa ottenere attraverso un veicolo tradizionale, ad un minor costo ed in maniera sicura. Essi possono essere impiegati per un gran varietà di operazioni e di applicazioni di pianificazione: individuazione degli incidenti, monitoraggio condizioni viabilità, coordinamento attraverso la rete di dispositivi stradali che dirigono il traffico, inseguimento di veicoli, misurazione dell'utilizzo di uno specifico tratto, stima dei flussi origine – destinazione.

Questi velivoli sono stati equipaggiati con una gran varietà di sensori prevalentemente di acquisizione immagine quali: visione notturna, ad infrarossi, multi ed iper spettrale, termali, laser scanner, sensori chimici e biologici.

In funzione al tipo di dato da campionare essi possono avere un sistema di trasmissione dati che permette di comunicare con la stazione base connessa via wireless, come nell'esempio in figura 2.3 nella pagina seguente, oppure un sistema di archiviazione dati on-board.

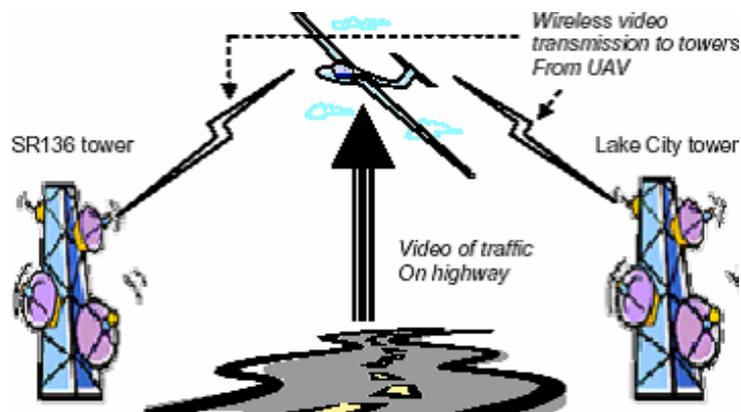


Figura 2.3: Esempio di architettura di scambio dati tra APR e stazioni base

Camcopter, indicato in figura 2.4 è uno dei tanti modelli realizzati dalla società Schiebel Corporation utilizzati a questo scopo. Esso è corredato da una base station che pianifica i dettagli di volo, immagazzina i dati e fornisce supporto al velivolo.

L'elicottero può essere programmato per volare autonomamente attraverso un'interfaccia grafica oppure manualmente. La tecnologia che consente il volo autonomo è basata sull'utilizzo dell'*Inertial Navigation System* (INS), accoppiata all'utilizzo di un sensore GPS, la quale assicura precisione della navigazione e stabilità durante tutta la fase di volo. Il computer di bordo può quindi salvare il percorso che seguirà autonomamente, se non modificato in tempo reale da un operatore operante sulla base station.



Figura 2.4: L'elicottero Camcopter

Metereologia

L'utilizzo degli APR nell'ambito meteorologico è essenziale: il fatto che un velivolo possa sorvolare una vasta zona in poco tempo consente di campionare una "fotografia atmosferica" delle condizioni presenti sopra il terreno analizzato difficilmente ottenibile con altri metodi.

La qualità dei dispositivi di campionamento montati a bordo di essi è inferiore a quella delle grosse stazioni di terra per ovvie ragioni di limitata massa trasportabile, tuttavia la facilità di gestione dei velivoli riguardo al lancio, controllo ed atterraggio è un punto a notevole favore del loro impiego così come il loro basso costo e la peculiarità di poter sorvolare zone vietate ad aereomobili con a bordo un pilota.

Nell'istituto di sistemi aereospaziali della Technical University di Braunschweig è stato sviluppato un mini APR capace di misurare con precisione le componenti tridimensionali del vento attraverso l'analisi del flusso d'aria attraversato, della temperatura, e dell'umidità [29]. E' stato inoltre implementato un pilota automatico che permettesse al velivolo di raggiungere zone al di fuori della copertura di controllo wireless, piuttosto limitata dovuta al basso costo. Non potendo quindi trasmettere in ogni zona sorvolata i dati in tempo reale, il dispositivo è stato dotato di una memoria a stato solido la quale, con soltanto 1 Watt di consumo energetico, permette il salvataggio dei dati relativi a svariate misurazioni. I sensori montati a bordo della piattaforma volante sono stati accuratamente selezionati per poter essere leggeri ed allo stesso tempo piuttosto precisi; la sonda dedicata all'analisi del flusso e delle turbolenze dell'aria è stata sviluppata appositamente dall'istituto di fluida meccanica dell'università di Braunschweig e presenta una sola massa di 9 grammi ed un diametro di 6 millimetri. E' stata inoltre previsto un utilizzo del velivolo a differenti quote, perciò ogni componente che andasse modificato e parametrizzato per la relativa altezza è stato progettato come componente removibile.

Per poter confermare la validità dei dati ottenuti a seguito di svariati campionamenti, è stato eseguito un confronto con gli stessi ottenuti da una sensoristica comune cablata ed agganciata ad un classico elicottero come da figura 2.5 nella pagina seguente.

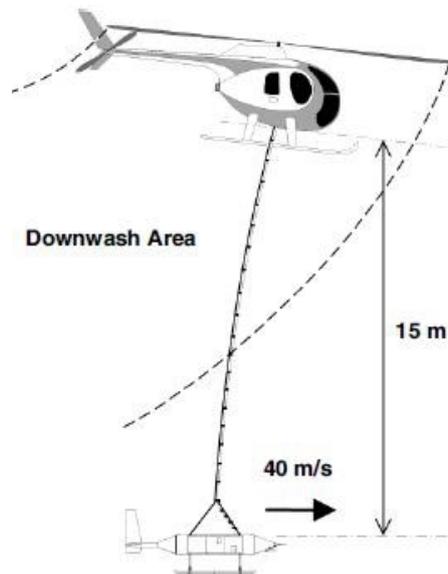


Figura 2.5: Classico sistema di campionamento dati metereologici

Considerando ininfluenza un basso margine di errore presente nelle statistiche, l'utilizzo di questo mini APR come nell'esempio in figura 2.6 fornisce dei dati estremamente validi permettendo una raccolta di informazioni con un dispositivo notevolmente più piccolo e pratico rispetto ai precedenti.



Figura 2.6: Esempio di mini APR

2.3.2 Natura ed agricoltura

Ottimizzazione piantagioni di caffè

Una delle principali limitazioni nell'utilizzo di velivoli che debbano sorvolare grandi aree è certamente la disponibilità a bordo di una grande quantità di fonte energetica.

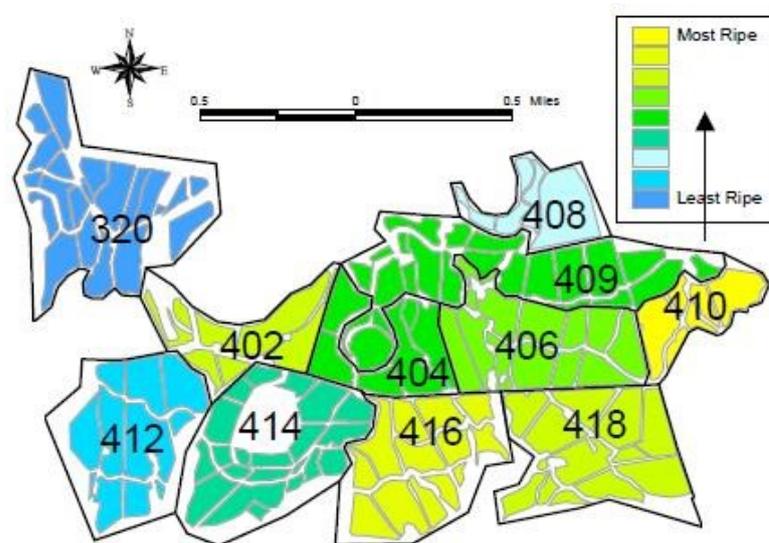


Figura 2.7: Mappa delle zone di coltivazione suddivise per grado di maturazione del prodotto

La Clark University in collaborazione con la NASA ha progettato un APR interamente alimentato da cellule fotovoltaiche posizionate a bordo [30].

L'obiettivo della realizzazione di questo prototipo chiamato *Pathfinder-plus* è stato quello di fornire informazioni dettagliate ai produttori di piantagioni di caffè quali il grado di maturazione del prodotto, la mancanza di irrigazione ed eventuali anomalie presenti nel terreno.

La zona comprendente una delle più grandi piantagioni di caffè presenti nel territorio americano situata sull'isola Kauai, parte delle Hawaii, è stata periodicamente sorvolata dal velivolo fornendo in ogni uscita della durata di 4 ore numerose immagini in alta risoluzione ottenute con l'utilizzo di due fotocamere dedicate.

Grazie all'utilizzo di una trasmissione wireless i dati ricavati in tempo reale sono stati costantemente elaborati ed hanno permesso la definizione di una mappa, presentata in figura 2.7 nella pagina precedente, nella quale i coltivatori potessero dedurre le zone nelle quali era fortemente consigliato effettuare il raccolto nel più breve tempo possibile. Anche in questo caso il dispositivo volante poteva essere controllato sia dalla stazione di terra sia dal proprio algoritmo installato a bordo.

Analisi della criosfera

Il processo di campionamento di immagini riguardanti ghiacciai è da molto tempo eseguito tramite l'utilizzo di immagini satellitari e di elicotteri dedicati. Entrambe le soluzioni, oltre ad essere piuttosto costose, non consentono rilevamenti topografici per ottenere un valido modello tridimensionale dell'area interessata. Nel sud dell'isola Bylot nel territorio del Canada è stato progettato dall'università di Calgary un velivolo autonomo, di supporto ad un elicottero pilotato, in grado di campionare immagini e dati utili alla ricostruzione digitale dell'area sorvolata [25].

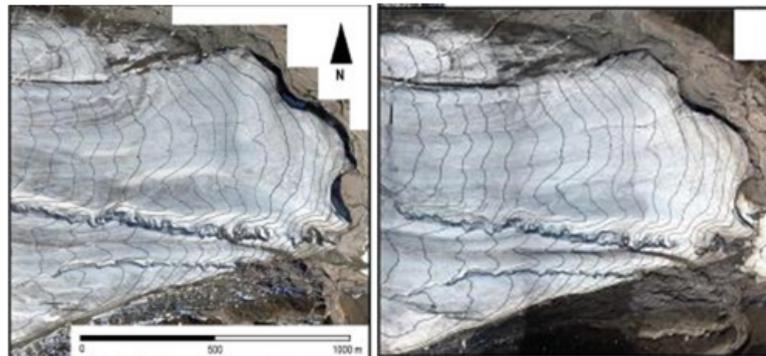


Figura 2.8: Analisi digitale del ghiacciaio effettuate a distanza di un anno

Questo modello non prevede una trasmissione senza fili dei dati campionati, perciò soltanto al termine del sorvolo è stato possibile analizzarli ed associarli a quelli dell'elicottero. Una fedele ricostruzione digitale dell'area interessata dal ghiaccio, confrontata con la stessa eseguita un anno dopo, ha potuto mettere in evidenza come da figura

2.8 nella pagina precedente il grave problema del restringimento dei ghiacciai.

2.3.3 Sicurezza

Pattugliamento di confine

In ambito marittimo attività quali il controllo dell'attracco navi, monitoraggio della merce scaricata ed identificazione del personale di bordo sono molto difficili da eseguire costantemente ed in maniera efficace. Sebbene una sorveglianza effettuata giorno e notte da velivoli pilotati possa contrastare svariati illeciti l'utilizzo di apparecchiature dedicate autonome, le quali permettano un controllo a 360 gradi, possono migliorare notevolmente la qualità della sicurezza ed ampliare le possibilità di rilevazione delle infrazioni. I principali vantaggi nell'utilizzo di APR in questo contesto sono innanzitutto un minor costo sia di costruzione che di gestione del velivolo associati ad una minor complessità di utilizzo, maggior affidabilità e meno danni subiti dalla struttura nell'eventualità di collisioni o cadute. Approvare il decollo di un'entità pilotata esclusivamente da remoto è notevolmente più immediato e la loro autonomia, grazie soprattutto al loro peso molto limitato, permette un'ampia copertura che può arrivare a qualche centinaio di km nel caso di modelli caratterizzati da piccole dimensioni.

Barnard Microsystems ha sviluppato a tal proposito un velivolo, presentato in figura 2.9 nella pagina seguente, che concentrasse il proprio design nella sicurezza, automazione e modularità [24]. Con un peso di soli 20kg esso è capace di trasportare una strumentazione con carico massimo di 4 kg e permette una prestazione massima della durata di volo pari a 7 ore. Pilotabile sia da un personal computer dedicato sia attraverso la sofisticata dotazione di bordo esso è versatile per qualsiasi tipo di requisiti richiesti e consente l'abilitazione di una navigazione interamente autonoma.



Figura 2.9: Il velivolo sviluppato da Barnard Microsystem

Localizzazione incendi boschivi

L'obiettivo del progetto COMETS, parte del programma IST della commissione europea, è stato quello di ottenere un coordinamento tra velivoli autonomi eterogenei per poter individuare zona colpite da incendi boschivi [27].

Sono stati progettati tre APR differenti per dimensioni e sensoristica:

- *L'elicottero Marvin* sviluppato dal gruppo Real-Time Systems and Robotics dell'università tecnica di Berlino, dotato di un sensore di calore e di una fotocamera digitale Reflex ad alta risoluzione.
- *Il dirigibile Karma* sviluppato dal laboratorio di analisi ed architettura dei sistemi a Toulouse, dotato di videocamere per la generazione di mappe in profondità.
- *L'elicottero Heliv*, presentato in figura 2.10 nella pagina successiva, sviluppato dal gruppo di robotica dell'università di Siviglia, risultato dell'evoluzione di un elicottero tradizionale dotato di funzionalità di sensing, percezione, comunicazione e controllo. Esso è dotato di videocamere visuali ed ad infrarossi.

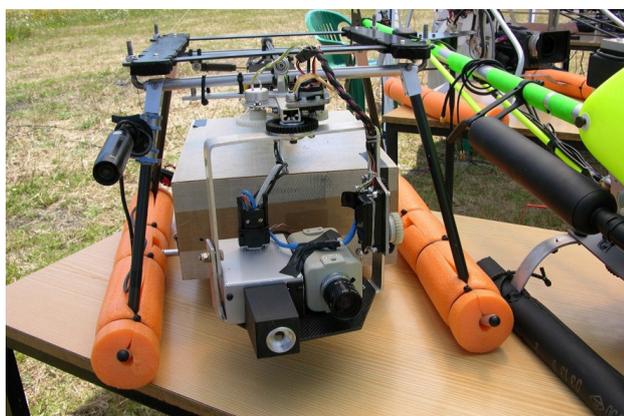


Figura 2.10: Videocamere visuali ed infrarosse montate sul velivolo Heliv

Ciascuna fonte di ripresa video è stata dotata della possibilità di movimento sui due assi, permettendo un cambiamento di visuale indipendente dalla posizione del velivolo in quota.

La prima fase delle attività svolte dall'architettura distribuita nel loro utilizzo è l'individuazione dell'incendio.

Funzionalità riguardanti l'analisi a basso livello delle immagini quali stabilizzazione, segmentazione come da esempio in figura 2.11 nella pagina seguente e geo referenziazione sono state definite per essere eseguite direttamente sulla piattaforma volante, riducendo notevolmente la banda altrimenti necessaria al trasferimento di grandi dati. Il loro risultato, oltre ad altri dati campionati da svariati sensori, viene condiviso in tempo reale tramite il *Black Board Communication System* (BBCS), protocollo di scambio dati basato sull'utilizzo di una memoria distribuita; La stazione base, parte fondamentale dell'architettura, confrontando i dati ottenuti dai velivoli in quota può così valutare la probabilità di un effettivo positivo riscontro riducendo la possibilità della presenza di un falso allarme.



Figura 2.11: Risultato di un'immagine segmentata

Non appena la zona interessata dall'incendio è definita inizia la fase di osservazione della zona colpita dalla calamità, ottenuta tramite l'hovoring dei velivoli, dalla quale si ottiene un flusso video stabilizzato che possa fornire dettagli utili all'estinzione del fuoco.

Capitolo 3

Tassonomia

Considerando la classificazione degli utilizzi di APR espressa nel secondo capitolo, il documento attraverso un'accurata analisi trasversale si propone ora di definire una possibile tassonomia delle funzionalità che un singolo APR possa offrire. Essa è redatta in funzione alle tipologie di informazioni necessarie ed ottenute, caratterizzanti ciascun possibile impiego di velivoli a controllo remoto ed indipendenti dal relativo contesto di applicazione. Per ciascun impiego diamo una descrizione degli obiettivi che si prefigge di raggiungere, fornendo svariati casi reali d'utilizzo che giustificano la sua presenza all'interno della tassonomia.

3.1 Campionamento

La necessità di dover sorvolare una specifica zona per poter campionare dati è molto frequente nei casi d'uso analizzati durante questo studio. L'obiettivo principale di questo impiego è la raccolta dati all'interno di un'area, tramite uno o più sensori di svariato tipo montati sul velivolo in azione, per poter effettuare statistiche, previsioni e ricostruzioni digitali. Nella maggioranza dei casi questi dati sono immagini relative a specifiche coordinate spaziali ottenute, ad esempio, in attività quali l'analisi della criosfera e la ricostruzione tridimensionale di un ghiacciaio. Tuttavia a seconda dell'ambito, essi possono spaziare da un campionamento di pressione ad uno di temperatura fino ad arrivare a complesse rilevazioni chimiche di gas e sostanze nocive.

3.2 Monitoraggio

Il monitoraggio non è nient'altro che un campionamento effettuato in maniera costante. Numerose sono le applicazioni che richiedono dati consistenti, costantemente aggiornati. A Bowie, in Maryland è stato realizzato il progetto Aereoview [1]: tramite l'utilizzo di fotografie ad infrarossi e la loro interpolazione in grafici è stato possibile dedurre le zone nelle quali le coltivazioni sono caratterizzate dalla presenza batteri. Il sistema inviando comandi opportuni, ha così potuto gestire intelligentemente il rilascio dei fertilizzanti portando ad un livello comune la crescita delle piantagioni. Sempre con l'utilizzo di immagini a colori e multispettrali ad alta risoluzione è stato possibile rilevare irregolarità nell'irrigazione nonché presenza di erbacce nei campi di caffè situati nelle isole Hawaii [20]. Analisi della qualità dell'aria, controllo della congestione del traffico stradale, controllo di attività vulcaniche ed analisi della composizione atmosferica sono altri importanti esempi di quest'impiego.

3.3 Ispezione

La più importante attività identificata nella stesura dell'elaborato che racchiude numerosi casi di studio riguardanti gli APR è senza dubbio la ricerca di anomalie eventualmente presenti in una determinata area, spesso connesse a fenomeni catastrofici o comunque pericolosi. Obiettivo di quest'impiego è quindi l'individuazione di uno o più riscontri ottenuti dall'analisi di campioni forniti dalla sensoristica del velivolo, i quali presentino un pattern congruente a quello considerato. I dati sono strettamente utilizzati per elaborazioni effettuate pressochè nell'immediato, sia a bordo sia tramite eventuali dispositivi collocati a terra. Oltre ad essere indispensabile per quanto riguarda il controllo in tempo reale del velivolo, la necessità di ottenere riscontri in un così breve periodo risiede nel fatto che la loro immediata individuazione permette una maggior efficienza nel compito in questione. Nel territorio americano tramite l'utilizzo di *LIDAR* (Laser Imaging Detection and Ranging), tecnica di telerilevamento che permette di determinare la distanza di un oggetto o di una superficie utilizzando un impulso laser, è stato possibile costruire una situazione tridimensionale del traffico stradale la quale ha messo in evidenza ingorghi e incidenti presenti nell'area sottoposta al controllo [6].

Sorvolando aree sovrastanti l’Azerbaijan, la Georgia e la Turchia è stato possibile individuare, seguendo il percorso di condutture marine adibite al trasporto di olio, zone nelle quali sono avvenute notevoli perdite. La sostanza fuoriuscita, avendo una temperatura differente rispetto a quella circostante poiché riscaldata dai raggi solari, è stata facilmente identificata tramite l’utilizzo di sensori termici [19].

Il DISAFRI, Dipartimento di Scienze dell’Ambiente Forestale e delle sue Risorse dell’Università della Tuscia di Viterbo ha da anni attivato una linea di ricerca sullo sviluppo di sistemi di telerilevamento aereo, dedicati allo studio delle risorse ambientali, agricole e forestali, caratterizzati da facilità di impiego, economicità e flessibilità d’uso [11]. La filosofia è quella di utilizzare vettori aerei leggeri sia monomotore che bimotori accoppiati a sensori progettati e sviluppati in un contesto di continuo progresso tecnologico.

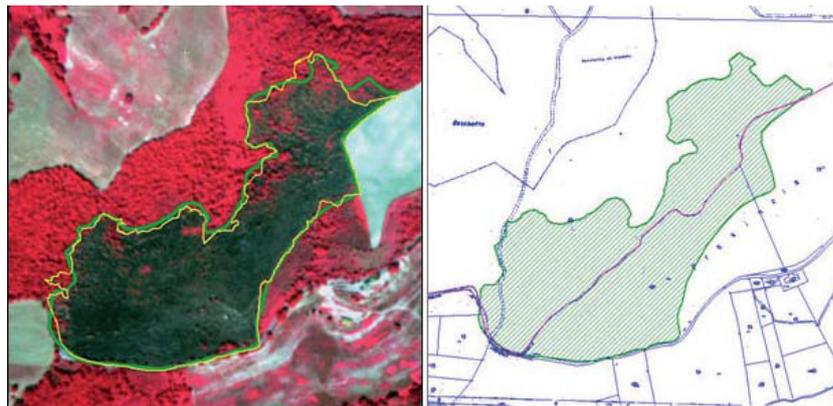


Figura 3.1: Incendio su ortofoto in falsi colori (in verde perimetro fotointerpretato, in giallo perimetro GPS) e relativa mappa.

Il sistema integrato di telerilevamento multispettrale utilizzato è composto da una camera a colori reali, una camera multispettrale a 3 bande (verde, rosso e vicino infrarosso), una camera termica, un GPS, un altimetro laser, ed un sistema di acquisizione dati a cui confluiscono tutti i flussi informativi nel quale vengono elaborati nel seguente modo: per ogni incendio si sono mediamente acquisite 1-3 immagini multispettrali con risoluzione media di 0,5 – 0,8 metri/pixel. Le immagini in falsi colori sono state infine elaborate attraverso le fasi di selezione, ortoproiezione e mosaicatura per ottenere un risultato come in figura 3.1.

Altri esempi sono il controllo dell'attraccaggio navi, la ricerca di persone disperse in territori montani soggetti a valanghe, l'identificazione di gas pericolosi e sostanze nocive presenti nell'aria, l'ispezione dei ponti, l'identificazione di rifiuti abusivi e l'individuazione di bracconieri.

3.4 Ricerca

Nel caso in cui non esista un pattern che possa racchiudere un insieme più o meno grande di possibili riscontri ma si consideri un attributo che identifichi un soggetto in maniera univoca, adottiamo allora il termine Ricerca come impiego affine all'individuazione di un'entità unica e distinguibile durante il sorvolamento di un'area.



Figura 3.2: Elementi di interesse presenti in una mappa utilizzata da archeologi

Uno dei principali lavori svolti dagli archeologi durante il ritrovamento di monumenti storici è la mappatura del terreno interessato a scavi. Nel passato attività svolte con il solo utilizzo di compasso, matite e fogli di carta oggi potrebbero vantare delle più svariate tecnologie moderne quali laser e computer dotati di un'elevata capacità di calcolo. Essendo tuttavia il costo di queste attrezzature molto elevato, impensabile da ricoprire con i pochi fondi a disposizione degli scienziati e presentando il loro utilizzo un notevole dispendio di tempo, esse non risultano essere la scelta ottimale. Ecco come

anche in questo caso, un'architettura basata sull'utilizzo degli APR possa portare notevoli vantaggi all'interno dell'usuale lavoro svolto dagli archeologi: contrassegnando ciascun monumento con un tag RFID e dotando uno o più velivoli con il relativo lettore di tag è possibile costruire una mappa piuttosto dettagliata dell'area di interesse senza che debba essere perlustrata fisicamente dagli archeologi. La mappatura può inoltre essere eseguita in due livelli: il primo, aereo, per la costruzione della mappa dall'alto; il secondo, a livello terreno, per ottenere una mappa navigabile intorno ad ogni singolo punto di interesse (vedi esempi in figura 3.2 nella pagina precedente) la quale, attraverso avanzate tecniche di elaborazione delle immagini, può rimuovere il paesaggio circostante mettendo in rilievo soltanto gli oggetti di interesse.

3.5 Inseguimento

L'inseguimento consiste nell'identificazione di un obiettivo da parte del velivolo e nel suo inseguimento fornendo il percorso effettuato ed eventuali altri dati campionati dai sensori. Nonostante si possano ottenere accurate angolazioni di visuale con i nuovi modelli di Action Camera presenti sul mercato, vi sono inquadrature impossibili da ottenere senza l'utilizzo di APR. Per questo motivo la società americana Falkor System ha deciso di investire nell'utilizzo di velivoli nel settore sport estremi. Un ulteriore esempio di questo impiego è il tracciamento del percorso di gru canadesi svolto dal governo americano. Tramite l'utilizzo di immagini è stato possibile registrare il percorso effettuato dagli animali in via d'estinzione per poter studiarne da vicino i loro comportamenti e spostamenti nonché effettuare il conteggio della loro numerosità [5].

Capitolo 4

Architettura

Nell'attuale capitolo proponiamo la procedura associata all'identificazione delle interfacce che il sistema si propone di offrire, derivata dalla classificazione degli impieghi ottenuta nel capitolo precedente, seguita dalla definizione di due possibili architetture idonee ad un'implementazione della relativa logica programmatica, necessaria all'utilizzo dei servizi.

Per ciascuna di esse analizziamo la composizione a livello logico dei livelli di deployment presenti valutandone vantaggi e svantaggi, identificandone il flusso di utilizzo da parte dell'utente e presentandone alcuni casi d'uso.

4.1 Interfacce a servizi

Per poter identificare le possibili interfacce a supporto degli impieghi definiti nella tassonomia presentata nel terzo capitolo, raggruppiamo dapprima i tipi di input e di output coinvolti in ciascun impiego ed in seguito, le loro combinazioni dedotte dalla classificazione presentata.

Un possibile raggruppamento delle informazioni in ingresso alle interfacce offerte dal sistema è composto dai seguenti gruppi di dati in input:

Geografici

Identificano una o più informazioni spaziali bidimensionali localizzate all'interno di un sistema di riferimento nelle quali il velivolo deve posizionarsi. Si trascurava la terza componente per un possibile movimento all'interno di uno spazio tridimensionale in quanto il sistema qui proposto rappresenta un *proof-of-concept* che può essere facilmente esteso. Tali informazioni possono essere:

Punti Insieme di punti, il cui ordine non vincola la loro visita.

Percorso Insieme di punti, il cui ordine vincola la loro visita.

Area Grandezza relativa all'area da sorvolare, entro la quale il velivolo deve attenersi.

Parametri

Dati che parametrizzano i servizi di monitoraggio ed inseguimento quali:

Numero di visite Numero di campionamenti da eseguire in ciascun punto durante l'esecuzione del servizio di monitoraggio.

Timeout Tempo dopo il quale il servizio di inseguimento si interrompe.

Servizi esterni

Ai fini di rendere l'utilizzo del sistema il più generico possibile per una sua massima integrazione all'interno di contesti di utilizzo differenti, è utile considerare elaborazioni strettamente legate all'ambito nel quale si opera delegate a servizi offerti da terze parti.

Tali servizi sono:

Visita Effettua il riordinamento di punti geografici. Utilizzato durante l'esecuzione dei servizi di Campionamento e Monitoraggio, attraverso una ottimizzata elaborazione, può ridurre il consumo energetico necessario al velivolo per poter raggiungere le zone desiderate.

Modellazione area Genera un insieme di punti geografici, appartenenti all'area definita dall'utente come parametro, a seguito di una chiamata dei servizi di Ispezione e Ricerca. Tali punti possono ad esempio rappresentare una griglia, una spirale oppure una sezione circolare all'interno dello spazio considerato.

Elaborazione dati Esegue elaborazioni più o meno complesse utilizzando le informazioni raccolte dalla sensoristica del velivolo. Tali dati, oltre ad essere consultabili da parte dell'utente nei servizi di Ispezione o Ricerca, possono essere indispensabili per definire la nuova posizione nella quale il velivolo deve spostarsi durante l'esecuzione del servizio di Inseguimento.

Le possibili informazioni che il sistema può fornire in risposta all'invocazione di un determinato servizio sono invece raggruppate come segue:

Dati Valori campionati dai sensori presenti nel velivolo. Nella maggior parte dei casi d'uso considerati nel secondo capitolo essi sono rappresentati da immagini.

Geografici Punto o punti nei quali il dato è stato riscontrato. Essi possono essere univoci quali longitudine o latitudine oppure relativi ad un sistema di riferimento definito ad-hoc.

Suddividiamo ulteriormente i valori in ingresso, in obbligatori ed opzionali per evidenziare quali di essi sono essenziali, inserendoli nella seguente tabella contenente i parametri necessari al funzionamento dei servizi presentati:

Impiego	Input obbligatori	Input opzionali	Output
Campionamento	<i>Geografici:</i> - Punti/Percorso	<i>Servizi esterni:</i> - Visita	<i>Dati:</i> - Immagini <i>Geografici:</i> - Punti
Monitoraggio	<i>Geografici:</i> - Punti/Percorso	<i>Parametri:</i> - Numero di visite <i>Servizi esterni:</i> - Visita	<i>Dati:</i> - Immagini <i>Geografici:</i> - Punti
Ispezione	<i>Geografici:</i> - Area <i>Servizi esterni:</i> - Modellazione area - Elaborazione dati		<i>Dati:</i> - Immagini <i>Geografici:</i> - Punti
Ricerca	<i>Geografici:</i> - Area <i>Servizi esterni:</i> - Modellazione area - Elaborazione dati		<i>Dati:</i> - Immagine <i>Geografici:</i> - Punto
Inseguimento	<i>Parametri:</i> - Timeout <i>Servizi esterni:</i> - Elaborazione dati	<i>Geografici:</i> - Area	<i>Geografici:</i> - Punti

Tabella 4.1: Tabella con impieghi e relativi input/output delle interfacce proposte

Osservando i prototipi ottenuti è possibile raggruppare differenti impieghi in un unico servizio, in quanto sono presenti interfacce con medesima semantica che differiscono nella sola definizione di alcuni di parametri.

L'architettura del sistema si identifica pertanto attraverso i seguenti tre servizi:

Sample Racchiude i servizi di Campionamento e di Monitoraggio.

Dati un insieme di punti, il servizio di Sample restituisce per ciascuno di essi uno o più valori campionati dal sensore. Il monitoraggio, ovvero un molteplici campionamento in ciascun punto, è contraddistinto dalla specifica del parametro corrispondente al numero di visite che il velivolo deve effettuare nei punti specificati.

Search Racchiude i servizi di Ispezione e di Ricerca.

Dati un'area, un servizio esterno di visita ed un servizio esterno di elaborazione dei dati ricevuti dal sensore il servizio di Search svolge un'ispezione che offre come risultato un insieme di punti con relative immagini campionate nei quali si è ottenuto un riscontro positivo da parte del servizio di elaborazione. La ricerca, ispezione che termina con il primo riscontro positivo ottenuto dal servizio di elaborazione dati, è ottenuta attraverso la presenza in fase di chiamata di un parametro dedicato.

Track Identifica il servizio di Inseguimento.

Dato un tempo massimo di inseguimento ed un servizio di elaborazione dei dati campionati che valuti il prossimo punto nel quale dirigersi, il servizio restituisce l'insieme di punti nei quali è stata riscontrata la presenza dell'oggetto inseguito.

In entrambe le proposte architetture individuate e presentate a breve, sono presenti tre principali livelli logici o tier i quali verranno esposti in diverse configurazioni di deployment:

Client Interfaccia utente relativa all'utilizzo dei servizi offerti dal sistema. Essa può essere fornita da un applicativo Desktop, un'applicazione mobile oppure un portale Web.

Elaborazione servizio Software relativo all'erogazione ed elaborazione dei servizi costituito dal server Web e dalle librerie relative al pilotaggio del velivolo.

Elaborazione dati Software relativo all'elaborazione dei dati ottenuti dai sensori a bordo del velivolo. E' strettamente legato al contesto di esecuzione dei servizi e racchiude tutte le possibili elaborazioni eseguite al termine dell'ottenimento dei dati forniti dal sistema. Esempi ne sono la modellazione tridimensionale di un ambiente, la creazione di planimetrie estese e la ricostruzione di complessi fenomeni atmosferici.

Si somma ad essi un ulteriore livello logico caratterizzato da un'interazione con servizi di terze parti descritti precedentemente; in tal modo è possibile disaccoppiare elaborazioni dei dati che possono variare a seconda dei requisiti di chi utilizza i servizi permettendo una più facile integrazione del sistema in un ecosistema più ampio. Esempi di questi servizi sono il riordinamento dei punti forniti al servizio di Sample, la generazione dei punti da visitare necessaria all'esecuzione del servizio di Search e l'elaborazione dei dati indispensabile a definire il prossimo spostamento del velivolo durante l'esecuzione del servizio di Track. Analizziamo in primo momento le possibili alternative di configurazione utilizzando il numero minimo di livelli di deployment pari a due, ipotizzando successivamente un'ulteriore interazione presente nel sistema descritta nella terza sezione di questo capitolo.

4.2 Due livelli di deployment

La più semplice architettura a supporto del sistema che ci proponiamo di realizzare è presentata in figura 4.1 ed è composta da due livelli di deployment, indispensabili al suo corretto funzionamento: il dispositivo nel quale è presente un interfaccia utente per l'utilizzo dei servizi offerti dal velivolo chiamato dispositivo di controllo ed il velivolo stesso. I vantaggi dell'adozione di questa proposta architetturale risiedono nella riusabilità dei servizi offerti dal sistema all'interno di svariati dispositivi client e nell'indipendenza della piattaforma di elaborazione dei servizi rispetto alla relativa implementazione. Ha senso considerare tale architettura ad esempio in scenari di utilizzo nei quali la stazione base può essere disconnessa momentaneamente dal velivolo.

I principali passi seguiti dal sistema per poter offrire i servizi sono:

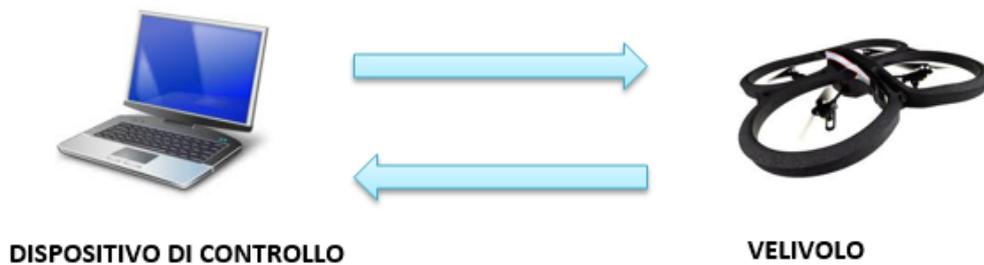


Figura 4.1: Architettura a due livelli di deployment

1. Invocazione servizio nel dispositivo di controllo con inserimento di parametri.
2. Elaborazione servizio invocato dall'utente.
3. Volo effettuato dal velivolo.
4. Elaborazione dei dati ottenuti dalla sensoristica a bordo del velivolo.
5. Visualizzazione dei risultati nel dispositivo di controllo.

Approfondiamone ora i principali descrivendo come le componenti software possano essere presenti in più configurazioni possibili, elencandone per ciascuna i relativi vantaggi e svantaggi.

4.2.1 Invocazione servizio

Azione obbligatoriamente eseguita sul dispositivo di controllo il quale può essere uno smartphone, un personal computer oppure una complessa stazione di terra. Essa deve presentare una semplice interazione con il sistema, nascondendo tutti i dettagli relativi alle interfacce presenti nei servizi e consentendo quindi un'agile azione da parte dell'utente.

4.2.2 Elaborazione servizio

Insieme di istruzioni che effettuano lo smistamento dei servizi ed impartiscono gli appropriati comandi al velivolo per poter ottenere i dati richiesti. Esso è strettamente dipendente dalla tecnologia con la quale i servizi sono offerti e dalle librerie utilizzate per il controllo dell'APR. La dislocazione di questo codice più ovvia è senz'altro, come da figura 4.2, nel dispositivo di controllo, poiché una sua implementazione a bordo della piattaforma volante potrebbe appesantire inutilmente il carico di lavoro già elevato del velivolo dovuto al controllo di tutte le variabili di volo.

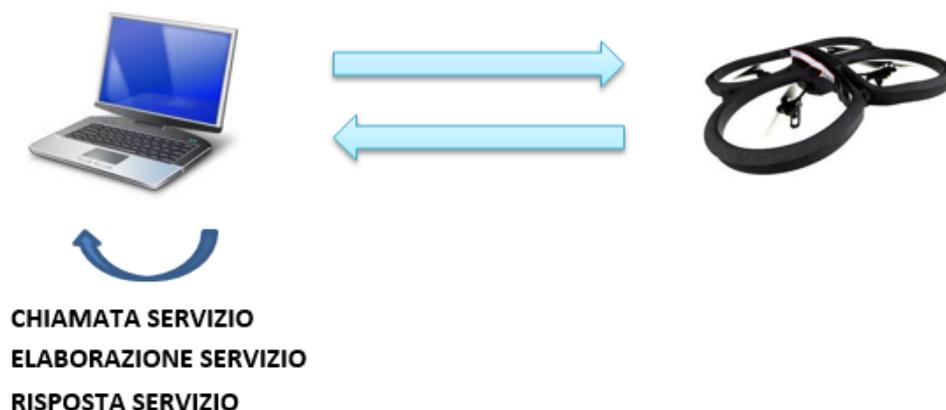


Figura 4.2: Architettura a due livelli di deployment con elaborazione del servizio eseguita nel dispositivo di controllo

Possono tuttavia presentarsi necessità d'uso, caratterizzate da un'elevata mobilità dell'utente all'interno dell'area sottoposta all'analisi del velivolo. Ne sono esempi l'inseguimento di un atleta durante un'esibizione, la ricerca di un disperso in territorio montano caratterizzato dalla presenza di valanghe

oppure l'individuazione di bracconieri; Prevedendo una configurazione nella quale sia il processo dedicato ai servizi sia la libreria di pilotaggio siano minimali, è possibile ipotizzare una completa gestione dei servizi offerti dalla componentistica a bordo dell'APR come da figura 4.3; dispositivi molto semplici, quali smartphone e tablet, riescono in tal modo a sostituirsi a stazioni base e computer portatili ottenendo una maggiore mobilità sul campo da parte dell'utente.



Figura 4.3: Architettura a due livelli di deployment con elaborazione del servizio eseguita nel velivolo

Differente sarà quindi l'interazione tra i due dispositivi rispetto al caso precedente: in questa configurazione sul canale di comunicazione saranno pertanto presenti soltanto le richieste dei servizi con le relative risposte, dalla quale si trae il vantaggio di un minor traffico dati impiegato.

4.2.3 Volo

Spostamento e campionamento nuovi dati effettuato dal velivolo. La trasmissione dei campioni può avvenire in tempo reale oppure in differita a seconda del servizio invocato: nel caso di Sample non si hanno requisiti real-time mentre nel caso di Search è utile ottenere dei riscontri nell'immediato per poter interagire tempestivamente.

4.2.4 Elaborazione dati

Codice relativo ad un'eventuale elaborazione dei dati campionati che comporta un calcolo computazionale maggiore rispetto alle altre fasi esposte. La miglior scelta per la maggioranza delle applicazioni analizzate è senza dubbio dislocarlo nel dispositivo sul quale risiede l'elaborazione del servizio come in figura 4.4: in primo piano esso è più idoneo a livello hardware ad eseguire operazioni sui dati e memorizzare i relativi risultati inoltre il risparmio di risorse presenti sul velivolo è fondamentale per poter raggiungere la migliore autonomia di volo.

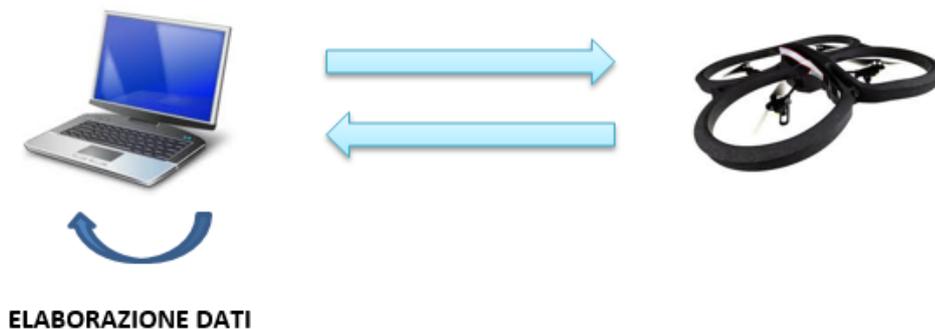


Figura 4.4: Architettura a due livelli di deployment con elaborazione dati eseguita nel dispositivo di controllo

Esempi che identificano meglio questa scelta sono il controllo di attività vulcaniche, l'analisi della composizione atmosferica oppure il controllo del traffico stradale: operazioni che richiedono l'analisi di una grande quantità di dati impensabile da ottenere in un dispositivo così limitato come un APR.

Una semplice elaborazione lato velivolo come da figura 4.5, indipendente o meno da un'ulteriore elaborazione lato utente, è utile progettartela soltanto per quei limitati casi d'uso caratterizzati da un semplice filtraggio delle informazioni.

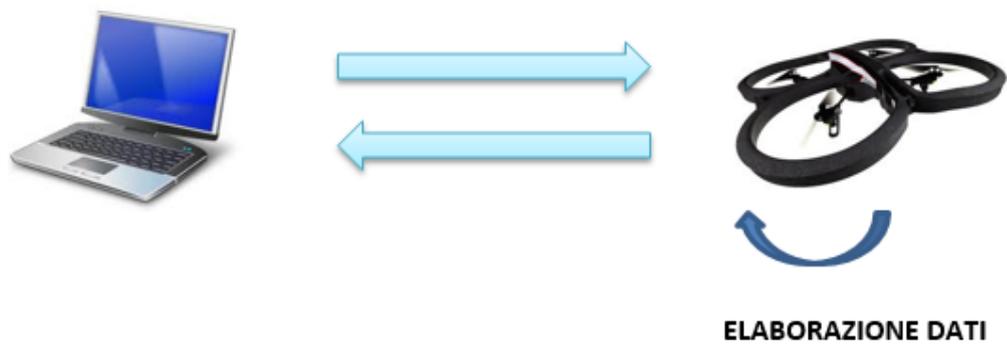


Figura 4.5: Architettura a due livelli di deployment con una semplice elaborazione dati eseguita nel velivolo

Ne sono esempi la mappatura di un sito archeologico ed il rilevamento di sostanze inquinanti ottenuto attraverso l'analisi cromatica di immagini campionate.

4.3 Tre livelli di deployment

A fronte dei casi d'uso considerati nel secondo capitolo nei quali è necessario poter utilizzare i servizi attraverso dispositivi minimali, arricchiamo ora l'architettura precedente con un terzo livello di interazione a supporto, rappresentato da una macchina server che alleggerisca il più possibile il carico di lavoro presente nel dispositivo di controllo, come indicato dalla figura 4.6. La sua presenza è inoltre necessaria per una maggiore integrazione delle funzionalità offerte dal sistema all'interno di processi di elaborazione scalabili e più complessi di un semplice applicativo client quali ad esempio Business Process e sistemi di Workflow Management. Tali processi spaziano dai Business Process [31] per un'integrazione delle funzionalità esposte dal sistema in ambito commerciale, ne è esempio l'utilizzo di APR come corrieri di consegna merce utilizzati da Amazon [2], fino ad arrivare a sistemi di Workflow Management [26] per una condivisione delle funzionalità esposte attraverso più gruppi di lavoro collaborativi, utile in una possibile gestione degli incendi come quella considerata nella terza sezione del secondo capitolo.



Figura 4.6: Architettura a tre livelli di deployment con elaborazioni servizio e dati eseguite da una macchina server

Il flusso base dei passi per poter utilizzare i servizi resta invariato rispetto all'architettura a due livelli di deployment, procediamo quindi ad analizzare soltanto le fasi che possono prevedere ulteriori configurazioni.

4.3.1 Elaborazione dei servizi

Assegnare alla macchina server la gestione dei servizi ha come vantaggio principale una possibile condivisione del velivolo tra più client. Utenti diversi possono richiedere servizi in contemporanea, gestiti sequenzialmente dalla macchina interposta in maniera trasparente ed ottimizzata. Essa potrebbe inoltre coordinare più velivoli, anche con framework e piattaforme diverse, per gestire casi d'uso piuttosto complessi come ad esempio la gestione di un incendio; durante la mappatura del sito è essenziale una forte cooperazione tra i diversi dispositivi che campionano ed analizzano il terreno affinché non si rianalizzino zone già sottoposte al controllo.

4.3.2 Elaborazione dei dati

La gestione dei dati dislocata sulla macchina server, oltre ad alleggerire il carico computazionale di client e velivolo, è utile considerarla dotata di un ampio spazio di memorizzazione nel quale possono essere salvate tutte le elaborazioni effettuate dal sistema. Questo archivio potrebbe inoltre essere fornito da un ambiente distribuito presente in un'architettura *Cloud*, aumentando l'efficienza nella condivisione e l'accessibilità delle informazioni eventualmente utilizzate da più processi. Grazie a questa scelta architetturale, nell'eventualità in cui un servizio dipendesse dall'elaborazione di dati precedentemente analizzati oppure richiedesse un servizio già invocato in precedenza in una finestra temporale di piccole dimensioni, il sistema può evitare di replicare un compito identico fornendo dati archiviati.

Capitolo 5

Implementazione REST

Descriviamo ora come le implementazioni delle interfacce identificate nel quarto capitolo e corrispondenti ai relativi impieghi catalogati nel terzo sono state realizzate.

Definiamo dapprima le implementazioni delle tecnologie considerate quali modello di APR con relativa libreria di utilizzo e tipologia di framework con relativo applicativo server per la distribuzione dei servizi basati sul Web. Segue l'elenco dei servizi terzi, utilizzati all'interno dei casi d'uso proposti nel sesto capitolo, necessari ad evidenziare l'elevata e semplice integrabilità del sistema in svariate necessità reali; E' punto chiave dell'elaborato sottolineare l'indipendenza dell'implementazione realizzata dal contesto di esecuzione ed utilizzo dei servizi.

5.1 A.R. Drone 2.0

Per poter ottenere dei risultati reali dall'utilizzo dei servizi che il sistema si propone di offrire è stato necessario identificare un velivolo economico e di piccole dimensioni, che potesse essere acquistato senza l'obbligo di dover conseguire licenze specifiche poichè destinato ad utilizzo strettamente amatoriale.



Figura 5.1: Schema dei componenti del Parrot AR Drone 2.0

La scelta è ricaduta sul modello di APR Parrot AR. Drone 2.0 [3].

Molto versatile e di utilizzo in ambiente sia interno che esterno, esso consente ad un vasto pubblico di divertirsi pilotandolo attraverso l'applicativo dedicato, supportato dai comuni sistemi operativi di smartphone presenti sul mercato quali Android ed IOS.

La tecnologia disponibile a bordo permette funzioni di controllo estremamente precise ed una buona stabilizzazione automatica.

Le principali caratteristiche, illustrate in parte in figura 5.1, sono:

- Struttura con tubi in fibra di carbonio.
- 4 motori senza spazzole a rotore interno dotati di CPU AVR a 8 MIPS.
- Processore ARM Cortex A8 da 1GHz 32bit, 1 GB di RAM e Linux 2.6.32.
- USB 2.0 Wi-Fi b,g,n.
- Giroscopio, accelerometro e magnetometro a 3 assi.
- Sensore di pressione e sensore a ultrasuoni per la misurazione dell'altitudine dal suolo.

- Telecamera HD 720p con obiettivo grandangolare e telecamera QVGA verticale per la misurazione della velocità in base al terreno entrambi con streaming a bassa latenza ed archiviazione diretta su USB oppure tramite Wifi sul dispositivo di controllo.

5.2 YaDrone

L'identificazione di una libreria da utilizzare che offrisse un'interfaccia semplice ed efficace per il controllo del velivolo è stata una delle attività più importanti per poter impostare al meglio, e nella maniera più concisa, lo sviluppo dei servizi. La libreria offerta dalla casa madre del prodotto, oltre ad essere scritta in un linguaggio di programmazione differente da quello scelto per la realizzazione del progetto, non presentava un'astrazione tale da poter pilotare il dispositivo attraverso semplici comandi direzionali. La ricerca del miglior framework da adottare è stata pertanto effettuata attraverso un'attenta e determinata consultazione dei portali Web riguardanti esperienze realizzate in precedenza quali progetti di ricerca e progetti open-source che ha portato a considerare le seguenti librerie:

JavaDrone

Nata con l'obiettivo di sostituire il codice nativo di controllo con Bytecode Java, questa libreria [12] è stata scartata poiché nonostante in alcuni forum fosse descritta utilizzabile anche con la seconda versione del velivolo, essa è compatibile soltanto con la prima versione del prodotto di casa Parrot.

ARDroneP5

Sviluppata da una studentessa asiatica, la libreria [4] presenta un'unica blanda documentazione di difficile comprensione comprendenti sezioni descritte perlopiù in giapponese. Dubbia è inoltre la compatibilità con la seconda versione del velivolo.

EasyDrone

Libreria [7] compatibile soltanto con la prima versione del velivolo.

YaDrone

Open framework scritto interamente in Java per il controllo dell'AR Drone 2.0 da remoto [21]. Con il suo utilizzo è possibile pilotare il velivolo interamente da PC senza dover appoggiarsi all'utilizzo dell'applicativo mobile.

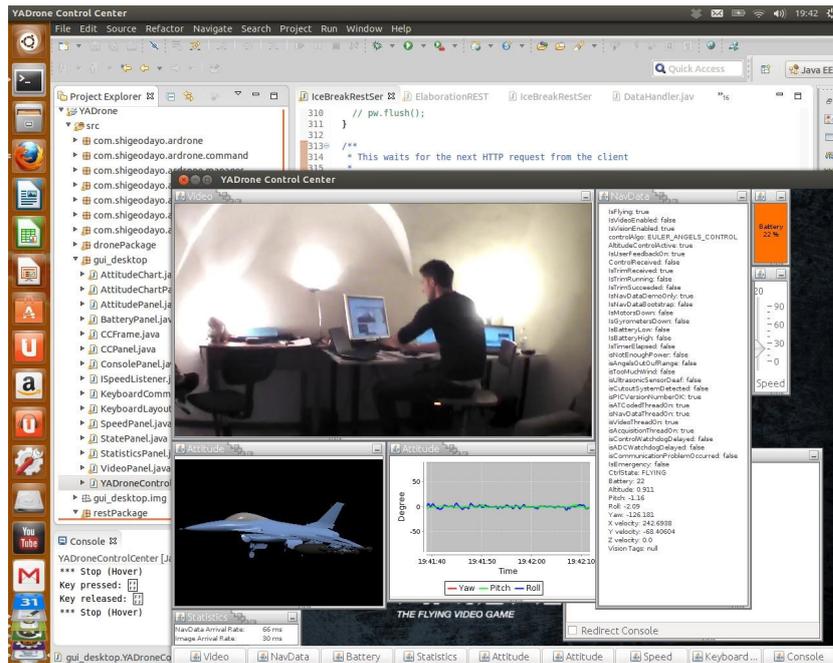


Figura 5.2: Esempio di utilizzo della classe Control Center presente nella libreria YaDrone

Esso presenta una pratica interfaccia, illustrata in figura 5.2, tramite la quale è possibile interagire fin dall'inizio con il dispositivo in maniera molto semplice ed intuitiva. Le principali componenti presenti in essa sono:

Streaming video Attraverso una classe Java dedicata all'interfaccia grafica, è fornito il flusso video che trasmettere le riprese effettuate dalla videocamera anteriore e da quella posta nella base del velivolo.

Indicatori giroscopio ed accelerometro Sono qui presenti tutti i dati ottenuti dai sensori quali giroscopio ed accelerometro, asso-

ciati ad una modello tridimensionale di aereomobile che rappresenta in tempo reale la posizione del APR nello spazio.

Indicatore batteria E' costantemente indicato il residuo di energia presente nella batteria a bordo.

Stato variabili di volo Sono presenti tutte le variabili con i relativi valori necessarie al corretto funzionamento del velivolo.

Console di debug Console che visualizza informazioni di debug.

5.3 IceBreakRestServer

Sebbene la dislocazione del codice relativo ai servizi possa apparire ininfluenza ai fini della realizzazione del progetto, essa è un punto chiave durante la definizione di un'architettura del sistema facilmente riconfigurabile. Nella maggior parte dei casi analizzati in precedenza il codice relativo al pilotaggio del velivolo risiedeva in un hardware dedicato, usualmente disposto a terra, indipendente dalla componentistica a bordo del velivolo. Attraverso la continua evoluzione della tecnologia e l'esponenziale aumento delle prestazioni che si possono ottenere in un dispositivo dedicato risulta utile, come più volte accennato all'interno del lavoro di tesi, considerare la possibilità di poter eseguire tutto il codice necessario al corretto pilotaggio di un velivolo a bordo di sé stesso.

Per poter offrire le funzionalità esposte adottando un'architettura leggera e comprensiva soltanto dell'indispensabile è stato quindi necessario individuare una tecnologia a servizi con relativo server Web minimale, sia come mole di codice, sia ottimizzato al risparmio energetico.

REpresentational State Transfer (REST) [17] è un tipo di architettura software per i sistemi di ipertesto distribuiti che meglio identifica la tecnologia a servizi necessaria agli scopi proposti: basata sull'utilizzo del protocollo HTTP, essa ridefinisce la semantica dei suoi comandi principali per poter offrire all'enorme numero di dispositivi che già lo supportano la possibilità dell'utilizzo di servizi Web.

Il campo di ricerca riguardante il server Web a suo supporto è risultato piuttosto ristretto in quanto è stato utile considerare Java, ovvero lo stesso linguaggio con il quale il framework di controllo del velivolo YaDrone è definito, come linguaggio di programmazione per una più semplice integrazione all'interno di un unico bundle.

Implementazione idonea ai nostri scopri è *IceBreakRestServer* [10]: un server REST scritto interamente in Java, il cui codice è caratterizzato da un basso consumo di risorse dovuto all'implementazione delle sole caratteristiche fondamentali ed indispensabili per un corretto funzionamento. Ecco un esempio del suo utilizzo che ne dimostra l'essenzialità:

Listing 5.1: Esempio di un utilizzo minimale del server REST IceBreakRestServer

```
import IceBreakRestServer.*;
import java.io.IOException;

public class Simple {

    public static void main(String[] args) {

        // Declare the IceBreak HTTP REST server class
        IceBreakRestServer rest;
        try {
            // Instantiate it once
            rest = new IceBreakRestServer();
            while (true) {
                // Now wait for any HTTP request
                // If we reach this point, we have received a request
                String name = rest.getQuery("name", "N/A");
                // we can now produce the response back to the client.
                // That might be XML, HTML, JSON or just plain text like
                here:
                rest.write("Hello world - the 'name' parameter is: " + name
                    );
            }
        }
        catch (IOException ex) {
            System.out.println(ex.getMessage());
        }
    }
}
```

5.4 Formato di ritorno dei dati

La risposta dei servizi utilizza come formato di interscambio dati il linguaggio di markup l'XML [8], acronimo di *eXtensible Markup Language*, basato su un meccanismo sintattico costituito dall'utilizzo di tag che consente di definire e controllare il significato degli elementi contenuti in un documento o in un testo.

L'elemento radice è, per ogni implementazione dei servizi proposti, il tag `<result>` che può contenere due possibili sotto elementi presenta in una o più occorrenze:

Point

Identifica un punto nello spazio. Contiene a sua volta i sotto elementi X e Y che identificano le coordinate del punto. Ecco un esempio codificato in XML:

Listing 5.2: Esempio della restituzione del punto (2.0;0.0)

```
<point>
  <x>2.0</x>
  <y>0.0</y>
</point>
```

Sample

Identifica un valore campionato dal sensore con associato il punto nel quale è stato acquisito. Ecco un esempio codificato in XML:

Listing 5.3: Esempio della restituzione del campione nel punto (2.0;0.0)

```
<sample>
  <point>
    <x>2.0</x>
    <y>0.0</y>
  </point>
  <value>
    data
  </value>
</sample>
```

5.5 Servizi esterni

Durante la progettazione dell'architettura, come anticipato nel capitolo precedente, è stato deciso di demandare l'esecuzione di processi quali la modellazione dell'area, l'elaborazione dei campioni ed il servizio di visita a servizi esterni per rendere il più possibile integrabile il sistema, in differenti contesti di utilizzo. Segue una breve descrizione di come essi sono stati realizzati all'interno del progetto:

Servizio di visita

Il servizio delegato a gestire la visita può riordinare, secondo un algoritmo definito dal programmatore, le coordinate ricevute permettendo di raggiungere un'ottimizzazione rispetto allo spazio che il velivolo deve sorvolare e quindi al relativo tempo impiegato per effettuare il campionamento.

Ai fini della nostra tesi questo servizio lascia inalterato l'ordine dei punti passatogli come parametro ed in seguito restituiti al chiamante.

Servizio di modellazione area

Fornendo in ingresso l'area nella quale il velivolo dovrà effettuare la ricerca è necessario generare dei punti per poter perlustrare l'intero spazio a disposizione. Il servizio di modellazione dell'area, utilizzando la metratura fornita come parametro, provvede a generare un insieme di punti nei quali il velivolo si sposterà in sequenza.

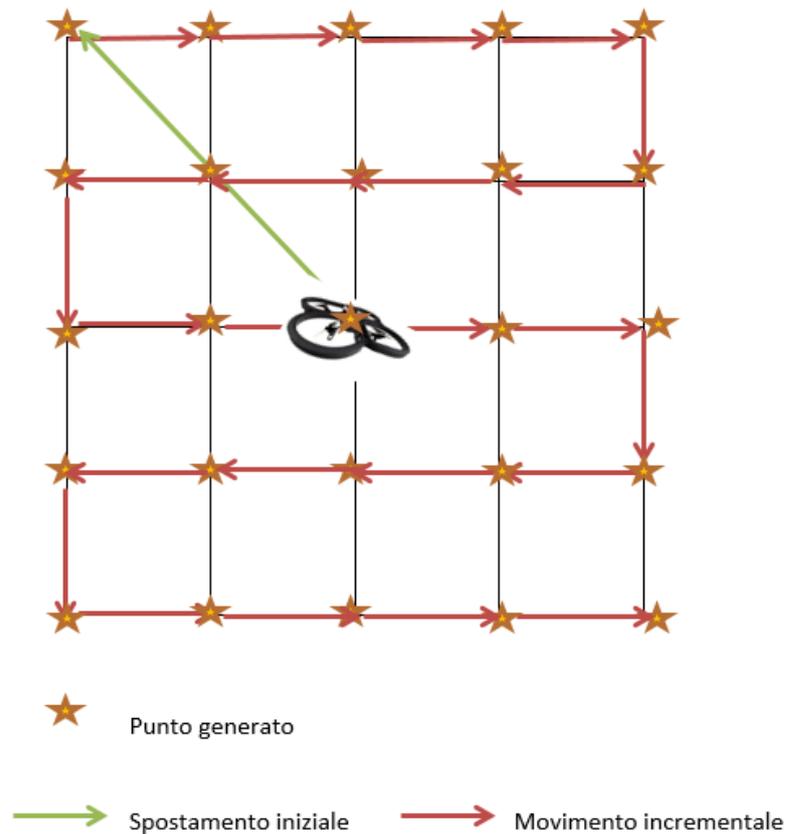


Figura 5.3: Esempio di utilizzo del servizio di modellazione area

Nella nostra tesi ipotizziamo che l'area fornita come parametro sia di forma quadrata ed il velivolo decolli dal centro dello spazio; il servizio di modellazione dell'area realizzato, dopo aver calcolato la misura di un lato, genera una griglia di punti da visitare con risoluzione pari ad un'unità come da figura 5.3. E' tuttavia possibile ridurre questa risoluzione considerando come risoluzione massima il un valore assunto dall'eventuale parametro $group$ indicato durante la chiamata del servizio. In figura 5.4 nella pagina seguente è presentata una griglia con risoluzione pari a 2 unità. L'ordinamento dei punti generati, indipendentemente dalla risoluzione della griglia impostata ed invertibile, prevede che la visita inizi dall'angolo in alto a sinistra aumentando ogni step la coordinata x di $group * 1$, rimbalzando continuamente tra i lati verticali. Ad ogni rimbalzo corrisponde un aumento della coordinata y

pari a group *1. Nel caso in cui il salto sia maggiore della dimensione rimanente della griglia, per non permettere la fuoriuscita del velivolo dall'area imposta, lo spostamento da effettuare viene arrotondato alla dimensione necessaria al raggiungimento del lato. La dimensione fisica di un'unità all'interno della griglia rispecchia i parametri di velocità e relativo tempo di spostamento del velivolo configurati all'interno della libreria *YaDrone*. Nel caso d'uso proposto tali valori sono stati settati in funzione allo spazio a disposizione.

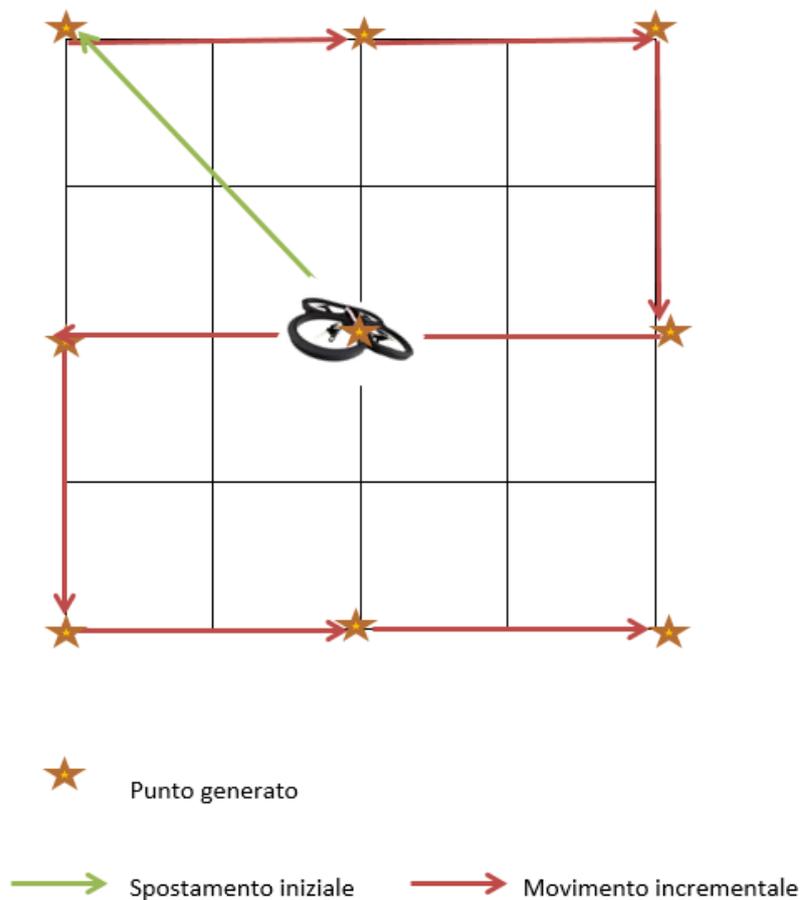


Figura 5.4: Esempio del servizio di modellazione area con raggruppamento punti pari a 2 unità

Servizio di elaborazione dati

Servizio dedicato all'elaborazione dei dati campionati dal velivolo.

Nella nostra tesi questo servizio restituisce una serie di dati preparati appositamente per l'esecuzione del caso d'uso presentato nel sesto capitolo.

5.6 Definizione servizi REST

La definizione delle interfacce REST descritta nelle seguenti tabelle, ognuna identificante un servizio, rispecchia i prototipi definiti nel capitolo quarto:

Tipo di Dati	Definizione
Parametri obbligatori	<ul style="list-style-type: none">• p<numero> = <coord.X> @ <coord.Y>
Parametri opzionali	<ul style="list-style-type: none">• times = <numero_visite>• reorderUrl= <URL_riordinamento_punti>
Oggetti di ritorno	<ul style="list-style-type: none">• <Sample>/ (<Sample>)+

Tabella 5.1: Tabella dei parametri, suddivisi per tipo di dato, definiti per l'utilizzo del servizio *Sample*

Tipo di Dati	Definizione
Parametri obbligatori	<ul style="list-style-type: none"> • area=<area> • locateUrl=<URL_visita_area> • elaborateUrl=<URL_elaborazione_dati>
Parametri opzionali	<ul style="list-style-type: none"> • search = true
Oggetti di ritorno	<ul style="list-style-type: none"> • <Sample> / (<Sample>)+

Tabella 5.2: Tabella dei parametri, suddivisi per tipo di dato, definiti per l'utilizzo del servizio Search

Descriviamo alcuni dettagli riguardo i parametri definiti:

- Numero progressivo: enumerazione dei punti
- Coord.X: numero che identifica la coordinata x del punto
- Coord.Y: numero che identifica la coordinata y del punto
- Numero visite: numero che identifica quante volte effettuare il campionamento nei punti passati come parametro
- Area: numero che identifica l'area in metri quadrati soggetta alla perlustrazione
- URL_visita_area: URL nel quale è raggiungibile il servizio assegnato alla gestione della visita, definito nella quinta sezione del capitolo quinto.

Tipo di Dati	Definizione
Parametri obbligatori	<ul style="list-style-type: none"> • time=<timeout> • elaborateUrl=<URL_elaborazione_dati>
Parametri opzionali	<ul style="list-style-type: none"> • area=<area>
Oggetti di ritorno	<ul style="list-style-type: none"> • (<Point>)+

Tabella 5.3: Tabella dei parametri, suddivisi per tipo di dato, definiti per l'utilizzo del servizio Track

- URL_riordinamento_punti: URL nel quale è raggiungibile il servizio assegnato al riordinamento dei punti, definito nella quinta sezione del capitolo quinto.
- URL_elaborazione_dati: URL nel quale è raggiungibile il servizio assegnato all'elaborazione dei dati, definito nella quinta sezione del capitolo quinto.
- Timeout: numero che identifica il tempo in secondi dopo il quale il sistema interrompe l'inseguimento dell'obiettivo.

Ciascun servizio è raggiungibile tramite una chiamata HTTP all'indirizzo così strutturato:

*http://<INDIRIZZO_IP_SRV_REST>:65000/
<NOME_SERVIZIO>?(<PARAMETRO>=<VALORE>)+*

5.7 Design programmatico

Il codice sviluppato per la realizzazione dei servizi Web è interamente scritto in Java. Le principali classi che consentono il corretto funzionamento del sistema sono:

DroneRESTServer.java Codice basato su *IceBreakRESTServer*, punto di accesso per l'utilizzo dei servizi Web. Identifica il server Web dedicato alla gestione delle richieste HTTP provenienti dalla rete. Esso provvede a smistare le informazioni ed attivare le componenti necessarie all'esecuzione del servizio invocato.

ElaborationRESTServer.java Codice che simula l'esecuzione dei servizi esterni presentati nel quarto capitolo e definiti in quello attuale. Essi sono identificati dalle seguenti implementazioni:

- *Reorder service*, corrispondente al servizio di visita.
- *Locate service*, corrispondente al servizio di modellazione dell'area nella quale si vuole ricercare oppure ispezionare.
- *Elaborate service*, corrispondente al servizio di elaborazione dati

Drone.java Codice basato su *YaDrone* che si interfaccia a livello UDP con il velivolo coordinandone il suo comportamento.

Sono inoltre presenti le seguenti classi di supporto:

DataHandler.java Codice che gestisce lo scambio di dati tra il velivolo e la parte di elaborazione.

InspectSender.java Thread lanciato a seguito di una chiamata al servizio Search, codice asincrono rispetto al continuo campionamento dei dati effettuato dal velivolo, delega l'elaborazione dei dati gestiti in una struttura condivisa al servizio di elaborazione e fornisce al client i riscontri individuati. Nel caso di impiego di ricerca, come descritto nel terzo capitolo, attraverso l'utilizzo di una struttura sincronizzata interposta tra esso e la classe principale a fronte del primo eventuale positivo riscontro nei dati campionati, provvede ad interrompere la visita del velivolo ed a notificare l'utente.

Sample.java Classe che identifica un campionamento, rispecchia l'omonima struttura XML.

Point.java Classe che identifica un punto nello spazio, rispecchia l'omonima struttura XML.

Utility.java Classe con metodi statici di supporto a funzionalità condivise.

Approfondiamo ora cosa accade a livello software per ciascun servizio offerto dal sistema.

5.7.1 Sample

Riprendendo l'architettura a due livelli di deployment, entriamo più nel dettaglio con uno schema delle interazioni tra le componenti software principali raffigurato in figura 5.5 nella pagina successiva ed il relativo Sequence Diagram presente in figura 5.6 nella pagina seguente.

Dopo aver ricevuto la richiesta di Sample il sistema esegue dapprima un eventuale riordinamento dei punti tramite il servizio terzo specificato come parametro (Passi 1 e 2 in figura 5.5 nella pagina successiva); In seguito invia tali punti, nel caso in cui il servizio sia un monitoraggio tante volte quante richieste, al velivolo (passo 3 in figura 5.5 nella pagina seguente) il quale per ciascuno di essi immagazzina il dato campionato dal sensore, associato alle relative coordinate, nella struttura dati predisposta (passo 4 in figura 5.5 nella pagina successiva). Al termine del campionamento il sistema preleva tutti i dati e li restituisce al client (passo 5 in figura 5.5 nella pagina seguente).

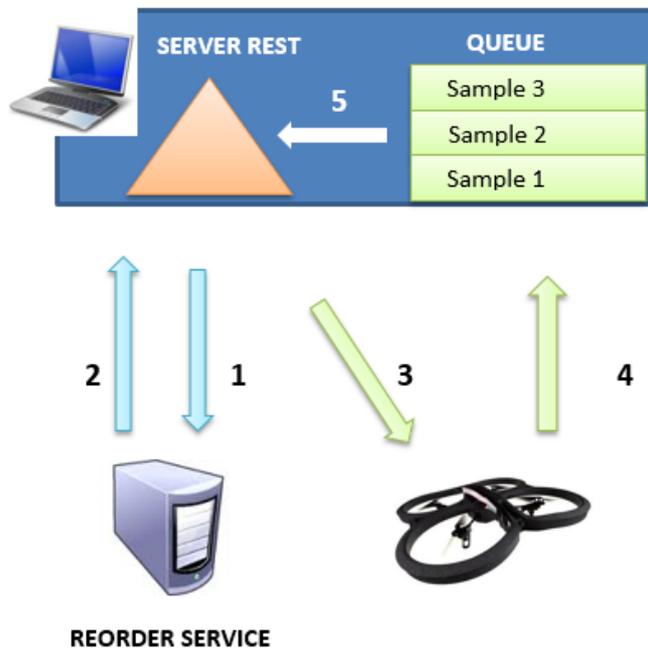


Figura 5.5: Interazioni componenti software del sistema nel caso di utilizzo del servizio Sample

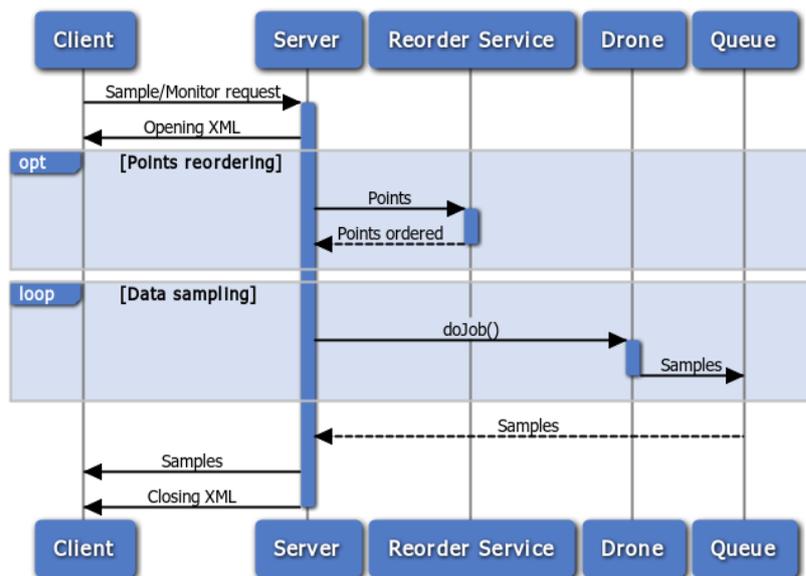


Figura 5.6: Sequence Diagram relativo all'utilizzo del servizio Sample

5.7.2 Search

Entriamo più nel dettaglio con uno schema delle interazioni tra le componenti software principali rappresentato in figura 5.7 nella pagina successiva ed il relativo Sequence Diagram presente in figura 5.8 nella pagina seguente.

Dopo aver ricevuto la richiesta di Search il server richiede dapprima la creazione dei punti tramite il servizio di modellazione area (passi 1 e 2 in figura 5.7 nella pagina successiva); in seguito esegue due processi in parallelo:

1. *Sampling data (Inspect sender thread)*: Il velivolo viene continuamente mobilitato per posizionarsi sui punti ricevuti, nei quali provvedere a campionare i valori dai sensori e quindi salvarli nella struttura dati condivisa (passi 3 e 4 in figura 5.7 nella pagina seguente).
2. *Elaborating data (classe RestServer)*: man mano che sono stati collezionati dati esso li preleva dalla struttura condivisa (passo 5 in figura 5.7 nella pagina successiva), li analizza interrogando il servizio di elaborazione dati (passo 6 e 7 in figura 5.7 nella pagina seguente) ed in caso di positivo riscontro restituisce al Client le coordinate nelle quali il dato è stato campionato associate alla relativa immagine.

Il sistema si interrompe con politiche diverse in funzione al servizio invocato: nel caso di Ricerca dopo il primo riscontro mentre nel caso di Ispezione dopo aver analizzato tutti i dati campionati.

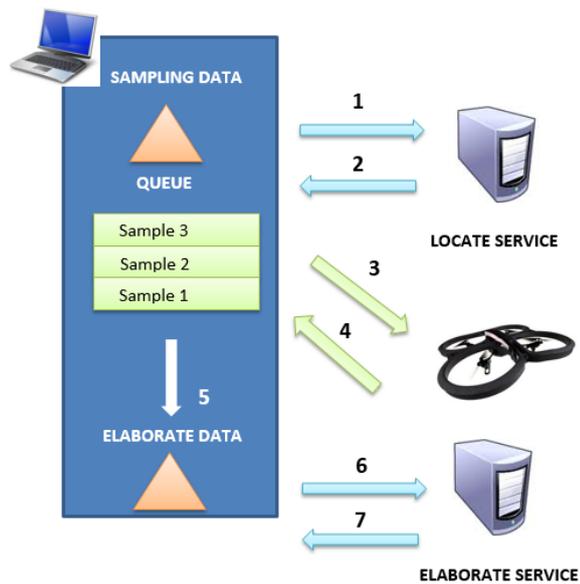
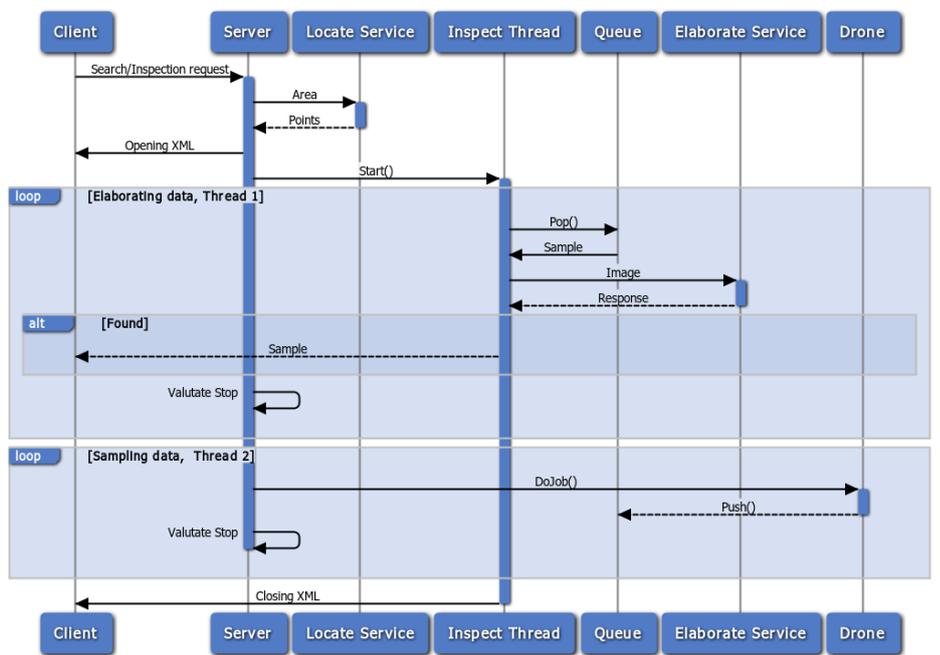


Figura 5.7: Interazioni componenti software del sistema nel caso di utilizzo del servizio Search



www.websequencediagrams.com

Figura 5.8: Sequence Diagram relativo all'utilizzo del servizio Search

5.7.3 Track

Definiamo lo schema delle interazioni tra le componenti software, appartenenti a questo servizio, presente in figura 5.9 seguito dal relativo Sequence Diagram raffigurato in figura 5.10 nella pagina seguente.

Il primo passo nella gestione del servizio di Track è l'avvio di un thread (*Timer thread*) che si comporti come timer in funzione del timeout fornito come parametro (passo 1 in figura 5.9). Fino al suo scadere (passo 6 in figura 5.9) il server eseguirà in ciclo le seguenti operazioni:

1. *Sampling data*: campionamento del campione trasmesso dal velivolo (passi 2 e 3 in figura 5.9).
2. *Elaborating data*: invocazione del servizio di elaborazione il quale analizza il dato e fornisce il prossimo punto nel quale dirigere il velivolo (passi 4 e 5 in figura 5.9).
3. *Response*: se il dato campionato identifica la presenza dell'oggetto da inseguire il server invia al Client la relativa coordinata

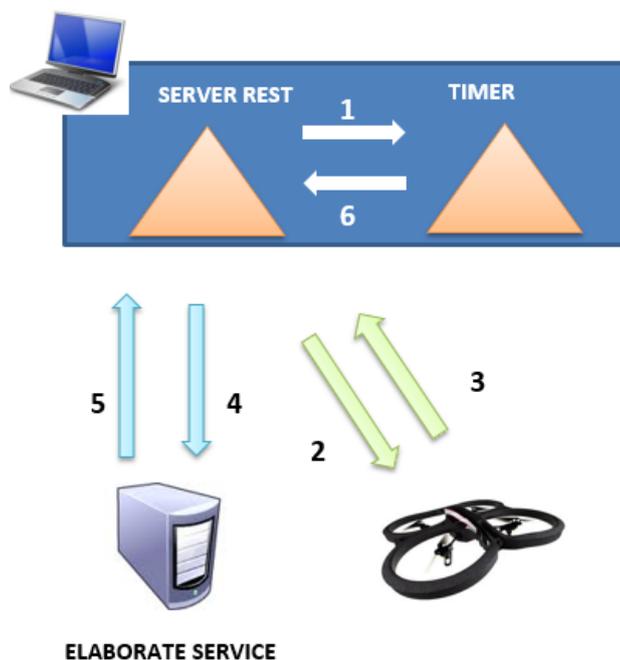


Figura 5.9: Interazioni componenti software del sistema nel caso di utilizzo del servizio Track

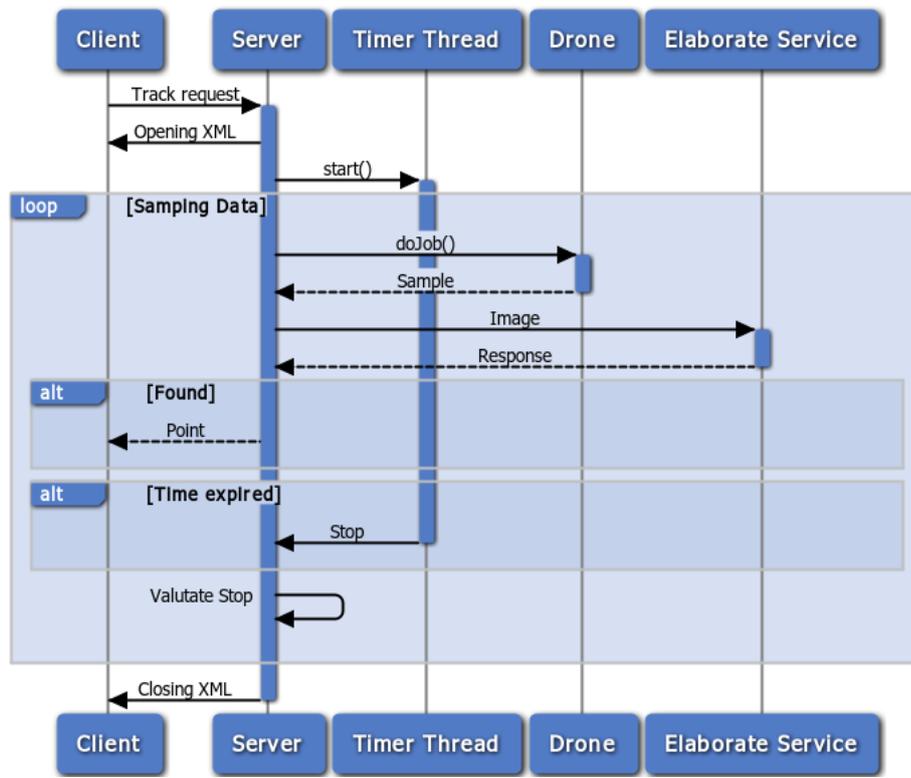


Figura 5.10: Sequence Diagram relativo all'utilizzo del servizio Track

Capitolo 6

Caso di studio

L'obiettivo, oggetto del lavoro in questione, è quello di fornire una definizione dei servizi Web che non si limiti ad una sola identificazione dell'architettura e delle componenti principali ma che proponga una valida implementazione di un caso di utilizzo reale definita da due esperimenti d'uso, ottenuta con il deployment del sistema in un Personal Computer associato all'utilizzo dell'APR presentato nel capitolo quinto. Tali esperimenti, relativi ai servizi di Sample e Search i quali raggruppano l'essenza del sistema, utilizzano come una delle possibili interfacce grafiche d'interazione al sistema il tool di Mashup *Social Mashlight*, lavoro di tesi sviluppato da Luca Chieppa realizzato in concomitanza alla tesi presente presentato ad inizio del capitolo.

Per ciascun di essi sono in seguito riportati il flusso di esecuzione che l'utente deve adottare, alcuni casi di studio descritti nel secondo capitolo che ne emulano l'utilizzo e le parti di codice sviluppate ad-hoc necessarie all'interazione tra i servizi Web ed il Mashup. E' infine presentato un sommario riguardante i risultati ottenuti da prestazioni misurate durante la loro effettiva esecuzione in ambiente reale, influenzata dall'inaffidabilità del canale di comunicazione wireless e da ulteriori fattori reali di applicazione quali presenza di vento e limitata autonomia di volo dell'APR, a dimostrazione della possibilità di adattare l'architettura ad una completa dislocazione dei servizi eseguita direttamente sul velivolo.

6.1 Web Meshup: Social Mashlight

Identificare una semplice ed intuitiva interfaccia utente necessaria all'interazione con il sistema è stato importante per dimostrare le potenzialità offerte dall'architettura. Applicazioni mobile, software desktop dedicati e complessi portali Web sono difficilmente integrabili con la molteplicità di servizi offerti tutt'oggi nel Web, oltre ad essere nativamente compatibili soltanto con una ridotta fascia di dispositivi disponibili ed utilizzabili da parte degli utenti.

Il concetto di *Mashup* [13] è nato negli anni 2000 come una prima risposta al nuovo paradigma *SaaS* (Software as a Service) [18]: dati eterogenei e provenienti da diverse fonti remote interagiscono tra di loro per raggiungere uno scopo comune, dando vita ad applicativi Web che diventano a loro volta nuovi servizi.

Un approccio di questo tipo comporta notevoli vantaggi riguardo le modalità di sviluppo di applicativi. Per uno sviluppatore è estremamente comodo ed efficiente poter attingere a software collaudato e sicuro senza un utilizzo eccessivo delle risorse interne.

La maggior parte degli strumenti messi a disposizione per la realizzazione dei *Mashup* fornisce un'astrazione di medio-basso livello, per un target utente esperto. Ne esistono alcuni che tentano di essere più semplici per coloro che non possiedono approfondite conoscenze tecniche, perdendo la flessibilità di utilizzo delle versioni più complesse.

Tra le diverse implementazioni di *mashup* presenti nella rete è stato dapprima identificato *Mashlight* come tool di riferimento, nato nel 2008 con l'obiettivo di fornire uno strumento versatile e leggero per lo sviluppo di applicazioni composte da servizi eterogenei.

Mashlight ha l'obiettivo di porsi trasversalmente rispetto alle possibilità offerte, ovvero permettere mashup complessi ma al contempo appoggiarsi ad un framework *lightweight* orientato ai processi.

Gli obiettivi di Mashlight sono:

Flessibilità Per supportare tutte le tipologie di Mashup in scenari differenti.

Usabilità Ovvero fornire ad ogni tipologia di utente un adeguato livello di astrazione.

Leggerezza e portabilità Per mettere a disposizione un ambiente leggero e compatibile con ogni dispositivo fisso o mobile.

Semplicità Per realizzare Mashup in modo rapido ed intuitivo.

Il vero elemento di novità che lo contraddistingue da altre implementazioni considerate è il concetto di *processo*, che permette di definire durante l'esecuzione quando i blocchi debbano essere eseguiti e quali dati devono utilizzare. Il processo viene definito mediante grafi orientati nei quali i nodi rappresentano le componenti funzionali, i blocchi appunto, mentre gli archi indicano le relazioni fra essi.

Social Mashlight, l'ultima evoluzione di questo applicativo, realizzata dal Dr. Luca Chieppa sotto la direzione del Prof. Luciano Baresi migliora ulteriormente l'architettura del precedente tool ottimizzando il carico di elaborazione lato utente e prestazioni di esecuzione.

La precedente versione era realizzata completamente mediante tecnologie Web, HTML JavaScript e XML, in quanto era concepito per un utilizzo locale client-side.

Questa versione invece è definita da un'elaborata architettura server basata sull'utilizzo di *Symfony*, diffuso framework PHP, che utilizza i concetti di classi come servizi e permette la suddivisione dei progetti in pacchetti modulari chiamati Bundle.

Questo ha permesso l'introduzione di nuovi standard di sicurezza nel codice, nonché una maggior velocità di esecuzione e una maggiore flessibilità nello sviluppo dei suoi componenti.

Il termine 'Social' nel nome sta ad indicare la nuova attenzione del framework verso le nuove realtà che si sono affermate negli ultimi anni: i Social Network. Sfruttare le ricchissime librerie messe a disposizione per esempio da Facebook, Google+ o Flickr, ha permesso a Mashlight di arricchirsi di nuove funzionalità, creando nuovi blocchi dalle capacità avanzate.

6.1.1 I componenti fondamentali

Si può suddividere il framework in componenti fondamentali, così da capire meglio come viene realizzato ed eseguito il Mashup.

Stickers

Il Mashup è composto da blocchi mobili che cooperano tra loro chiamati Stickers. Sono avvolti da una cornice blu e identificati da un proprio nome specifico. Possono essere trascinati liberamente all'interno della parte centrale della pagina e si può agire anche sulle loro dimensioni.

Ogni Sticker ha una propria funzionalità specifica: si possono considerare come delle mini applicazioni Web 2.0 indipendenti tra loro, con una serie di parametri di ingresso e di uscita e funzioni di *update*.

Esempi di Sticker possono essere blocchi che visualizzano mappe, calendari o strutture più complicate come dei form di inserimento dati. Quasi tutto può diventare uno Sticker con opportuni accorgimenti nella loro programmazione.

Ognuno ha un proprio set di risorse: pagine e template HTML, CSS e JS, immagini e icone che devono essere poste all'interno di una cartella specifica che deve seguire una nomenclatura formata da [Produttore][NomeSticker].

È inoltre presente un file XML di manifesto, che ha lo scopo di indicizzare tutte le risorse interne e assegnarne dei parametri formali, come per esempio nome e dimensioni. In questo file sono anche elencate le funzioni di ingresso e uscita dell'applicazione, e altre funzioni utili al caricamento iniziale o al suo aggiornamento.

Griglie

In Social Mashlight una griglia è l'equivalente di una singola schermata, in cui gli Stickers collaborano tra di loro senza dover aggiornare la pagina. Possiamo definirla come un *macroblocco*, composizione di blocchetti più piccoli, ma anch'essa con parametri di ingresso e di uscita.

Quando una griglia termina la sua esecuzione, Social Mashlight salva tutte le variabili di uscita e le passa come parametri in ingresso alla griglia successiva. Questo avviene mediante l'utilizzo delle sessioni PHP.

Ricapitolando, ad ogni griglia quindi deve corrispondere almeno uno Sticker; l'insieme di tutte le griglie compone il processo completo del Mashup.

Il Processo

I blocchi vengono collegati tra loro attraverso le connessioni di input ed output, definite come in figura 6.1, allo scopo di formare un flusso detto *flusso di processo*, rappresentabile mediante un grafo orientato illustrato in figura 6.2 nella pagina seguente. Tale grafo è la somma di due grafi specifici: *flusso di esecuzione* e il *flusso dati*.

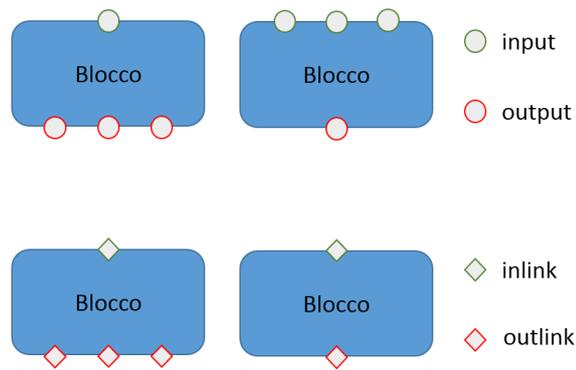


Figura 6.1: Esempio relativo a punti di connessione di blocchi definiti in Social Mashlight.

Il *flusso di esecuzione* è il cammino percorso dall'utente nella sua interazione con i blocchi. Esso è dotato di un unico nodo iniziale e finale. Sono ammessi cicli, biforcazioni ed unioni.

Il *flusso dati* invece mostra lo scambio di dati tra i blocchi nell'esecuzione del processo. Il passaggio di parametri può avvenire per assegnamento di un valore in uscita da un blocco precedente o di un valore costante.

L'unico vincolo è che l'assegnamento dei parametri sia consistente con il flusso del processo, in quanto la disponibilità del valore del parametro di uscita del blocco è disponibile dopo la sua esecuzione.

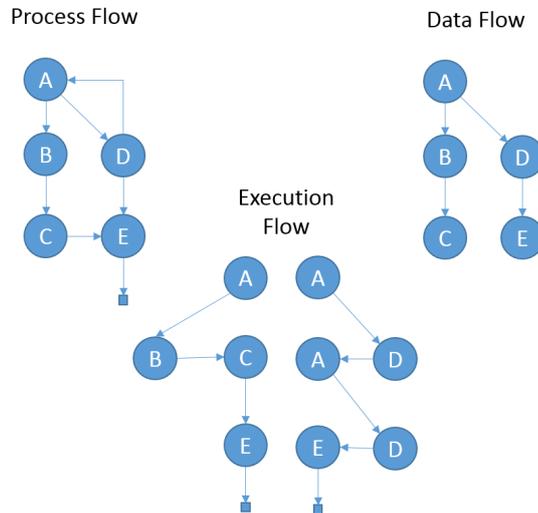


Figura 6.2: Esempio di flusso di processo valido.

Il processo viene quindi elaborato a partire da un file XML, denominato *manifesto di processo*, che il framework traduce nell'esecuzione del Mashup.

6.1.2 L'architettura

Social Mashlight è suddiviso in tre diversi bundle che svolgono ruoli diversi: presentazione, logica, dati.

GridmeBundle

GridMeBundle si occupa di costruire il sito Web che ospita al proprio interno i Mashup, e quindi ricopre tutte quelle funzionalità volte alla presentazione.

Si è deciso di creare un bundle a parte per rendere indipendente la presentazione dal motore interno. In questo modo si può in futuro modificare la piattaforma con cui si utilizza Social Mashlight, oppure affiancare a quella attuale una rappresentazione grafica ottimizzata per tablet e dispositivi mobili.

In figura 6.3 nella pagina successiva si possono notare cinque diverse zone, le quali possono essere ridimensionate o nascoste a piacimento. Ognuna svolge un proprio compito ben preciso, e in questa trattazione useremo le coordinate cardinali per identificarli in modo preciso.

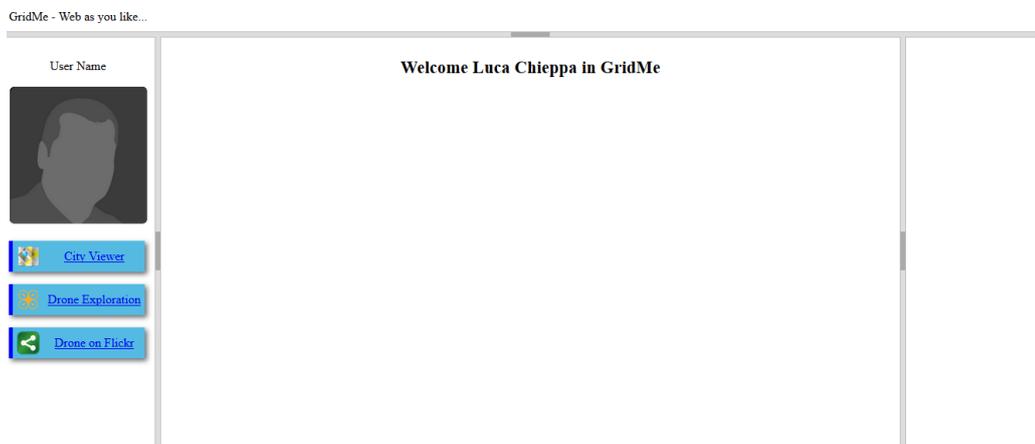


Figura 6.3: GridMeBundle.

Il pannello nord e il pannello sud si occupano di mostrare le informazioni base sulla piattaforma come il logo e il copyright.

Il pannello ovest ha la funzione di mostrare le informazioni dell'utente, come il suo nome e la foto. Inoltre ha lo scopo di visualizzare tutti i Mashup realizzati dall'utente come un elenco ordinato.

Ogni voce viene contraddistinta da un nome e da un'icona, assegnati durante la sua realizzazione per permettere una miglior distinzione. Cliccando su una delle voci si comanda al framework di richiamare il relativo Mashup.

Questo pannello, con quello nord e sud, non cambierà il suo contenuto durante l'utilizzo della piattaforma.

La vera parte vitale del framework è il pannello centrale. Esso ha lo scopo di visualizzare una schermata di ingresso e, al caricamento dei Mashup, di visualizzare la griglia corrente. L'utente potrà interagire con ogni componente posto al suo interno, procedendo nell'esecuzione del processo oppure modificare posizione e dimensione degli Sticker a proprio piacimento.

L'ultimo pannello è quello posizionato a est. Tipicamente risulterà 'collapsato' all'avvio di un mashup, in quanto ha una funzione meramente tecnica. Il suo scopo è quello di mostrare all'utente la struttura del processo, ovvero come gli Sticker interagiscono tra di loro tramite una struttura a nodi.

MashlightBundle

Il vero cuore del framework è composto dalle componenti di questo bundle, che si occupa dello strato logico dell'applicazione.

È strutturato in diversi servizi che si occupano di concorrere nella creazione dei contenuti che verranno utilizzati dal GridMeBundle per la visualizzazione della pagina all'utente.

È qui che si realizza la costruzione del Mashup e la sua gestione.

Il servizio CoreEngine si occupa di:

- Richiamare l'elenco dei Mashup disponibili per l'utente.
- Leggere la struttura del processo e creare una rappresentazione interna.
- Mantenere in memoria la struttura dei nodi e delle griglie.
- Fornire la griglia corrente da visualizzare all'utente.
- Gestire i meccanismi di autenticazione e comunicazione con i database e le librerie.

Per costruire il processo, MashlightBundle:

1. Legge da un file XML la struttura del processo e l'elenco dei moduli da caricare.
2. Per ogni modulo, legge il relativo file di manifesto dello Sticker e importa tutte le risorse disponibili.
3. Crea la struttura del processo tramite alberi interni e strutture ad array e classi.
4. Produce in output una serie di blocchi di codice, suddivisi per risorsa (CSS links, JS links, HTML Code).
5. Produce in output il codice JS che viene utilizzato per l'esecuzione del processo.

Sarà compito del GridMeBundle utilizzare queste risorse per la composizione della pagina.

StickersBundle

Questo bundle contiene tutti gli Stickers che Social Mashlight mette a disposizione per la creazione dei Mashup.

Differentemente dagli altri due bundle del framework non si occupa della loro visualizzazione.

Possiamo considerarla come una libreria di componenti, che gli altri bundle usano per costruire i Mashup, oppure come un ‘magazzino’ di risorse.

Gli Stickers presenti in questa sezione sono ‘inerti’: finché non vengono elaborati dal framework, sono un mero insieme di file e immagini, senza la possibilità di portare a termine alcun compito.

Social Mashlight si occuperà di contestualizzarli e dargli una ‘vita’: verrà assegnato un identificativo univoco allo Sticker, verranno importate le risorse disponibili, e si metteranno in collegamento le sue funzioni con la griglia che lo ospita. Da quel momento il blocco potrà essere parte del processo.

La flessibilità del framework sta proprio in questo: dallo stesso asset di risorse, è possibile creare diversi Stickers, che a seconda del processo in cui sono inseriti possono avere comportamenti anche molto diversi tra loro.

Lo Sticker può essere realizzato sfruttando qualsiasi tipo di risorsa di tipo Web, quindi codice Javascript, HTML, XML, immagini, fogli di stile etc.

L’unica risorsa obbligatoria è la presenza di un particolare file XML che è il manifesto dello Sticker.

Listing 6.1: Esempio di file config.xml

```
<?xml version="1.0" encoding="UTF-8"?>

<block>
  <about>
    <name>Image Viewer</name>
    <icons>
      <icon size="128">images/map-icon.png</icon>
    </icons>
    <folder>lukiep-imageviewer</folder>
    <style>
      <size width="680" height="450" />
    </style>
  </about>

  <components>
    <javascript-urls>
      <javascript-url
        type="relative">js/show-image.js</javascript-url>
    </javascript-urls>

    <css-urls>
      <css-url>css/viewerstyle.css</css-url>
    </css-urls>

    <div-ids></div-ids>

    <html-contents></html-contents>

    <twig-files>
      <twig-file>Lukiep:image-viewer.html.twig</twig-file>
    </twig-files>
  </components>

  <functions>
    <inlink>
      <function name="loadImages">
        <variable type="string" name="stringimage"
          isArray="false"></variable>
      </function>
    </inlink>
  </functions>
</block>
```

```

        </function>
    </inlink>

    <outlink></outlink>

    <onload>
        <function name="showPreloadFun"></function>
    </onload>

    <variables></variables>

    <update>
        <function name="loadImages">
            <variable type="string" name="stringimage"
                isArray="false"></variable>
        </function>
    </update>
</functions>
</block>

```

Come si vede dall'esempio di listato sovrastante, esso è suddiviso in diversi tipi di descrittori:

- Una serie di campi che descrivono il blocco, come il nome e l'autore.
- Alcuni parametri come `<folder>` che vengono utilizzati per localizzare le risorse.
- Una serie di parametri che istruiscono il framework su come il blocco debba essere visualizzato.
- L'elenco delle risorse dello Sticker.
- L'indicizzazione di tutte le funzioni JavaScript, e il loro ruolo nell'esecuzione del blocco.

All'interno del tag `<components>` le risorse vengono raggruppate in base alla loro tipologia.

Elenchiamo di seguito le realizzazioni di Stickers presenti nativamente all'interno di *Social Mashlight*:

Image Viewer



Figura 6.4: Sticker dedicato alla gestione di una galleria di immagini.

Il blocco in Figura 6.4 è uno Sticker interno a Social Mashlight che si occupa di ricevere un insieme di immagini e di visualizzare una galleria.

Flickr



Figura 6.5: Sticker dedicato al caricamento remoto delle immagini sul portale Web Flickr

Lo Sticker mostrato in Figura 6.5 consente, attraverso la pressione sul pulsante "Send to Flickr", una semplice ed immediata condivisione delle immagini, caricate dal sistema sul famoso portale Web dedicato alla gestione delle immagini.

6.2 Integrazione in Social Mashlight

Tramite le funzionalità messe a disposizione da *Social Mashlight* si sono potute creare interfacce grafiche come possibile integrazione delle funzionalità offerte dal sistema proposto, adibite al controllo di un APR attraverso i servizi esposti.

Presentiamo ora gli Stickers realizzati suddivisi per servizio Web offerto:

6.2.1 Sample

Obiettivo di questo esperimento è quello di consentire all'utente di usufruire del servizio Sample, descritto nella prima sezione del quarto capitolo, con il quale è possibile visualizzare immagini rilevate in determinati punti impostati a priori. Definiamo gli strumenti che permettono all'utente di poter selezionare dei punti all'interno dello spazio nei quali commissionare all'APR il campionamento di immagini ed in seguito visualizzare il risultato.

Drone form - Prima scheda

In figura 6.6 nella pagina seguente è mostrato il componente tramite il quale è possibile fornire all'APR le coordinate ed i dati necessari, come definiti in tabella 5.1 a pagina 48, con i quali effettuare le proprie rilevazioni.

Ogni punto sulla griglia rappresenta una coordinata in un'area di ricerca definita a priori. Selezionando un punto si comanda al velivolo di effettuare una rilevazione in quella determinata coordinata, producendo una foto.

Drone Form

Sample / Monitor Search / Inspect Track

Numero di campionamenti 1

URL servizio riordinamento punti http://localhost:65001

S

Reset Proseguì

Punti selezionati 0

Figura 6.6: Sticker dedicato al controllo dell'APR per le sue funzionalità.

Il punto centrale, marcato con l'etichetta 'S' per 'Start', è la coordinata di riferimento da cui parte l'APR. Tutti gli altri punti rappresentano le coordinate relative a partire da quella di origine.

Tramite lo spinner 'Numero di Campionamenti' si comanda all'APR quanti rilevamenti fare su ogni coordinata.

Il form 'URL servizio riordinamento punti' invece è un indirizzo che viene fornito al servizio di gestione REST di gestione dell'APR, per riordinare le coordinate selezionate in modo da poter creare un percorso ottimizzato.

Drone form - Viewer

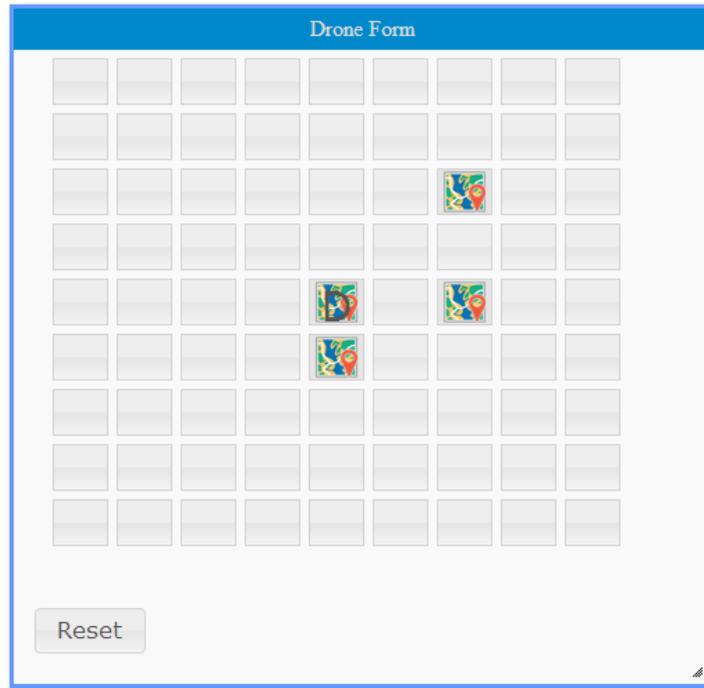


Figura 6.7: Sticker dedicato alla visualizzazione dei risultati dei servizi di Sample, Search e Track all'interno di una griglia

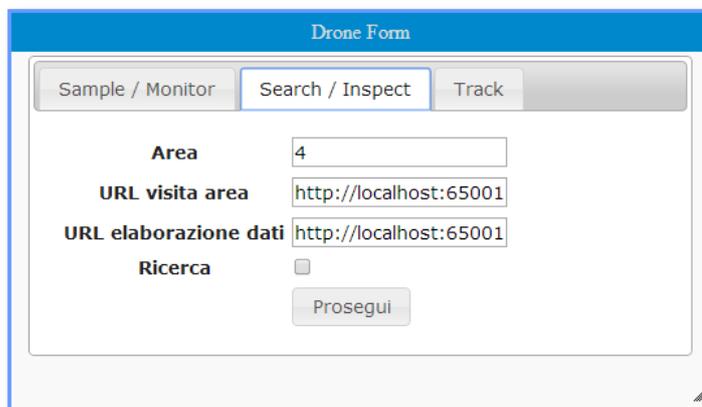
Il blocco in figura 6.7 ha la stessa disposizione di pulsanti come in Drone Form in figura 6.6 nella pagina precedente, disposti in una griglia 9x9. Su questa griglia non è però possibile impartire istruzioni al velivolo, ma è utilizzata per la presentazione dei risultati. I punti di interesse rilevati dall'APR identificati attraverso l'esecuzione del servizio vengono contraddistinti da un'icona che rappresenta una mappa. Nonostante si fosse potuto utilizzare il medesimo Sticker precedente per impostare i punti e visualizzarne le relative immagini, si è deciso di separare le funzionalità in due blocchi distinti per una migliore distinzione logica dei contenuti e per rimuovere informazioni inutili visualizzate al termine dell'invocazione del servizio.

6.2.2 Search

Obiettivo di questo esperimento è quello di consentire all'utente di usufruire del servizio Search, offerto dal sistema e descritto nella prima sezione del quarto capitolo, con il quale è possibile ricercare pattern o oggetti all'interno di un'area e visualizzarne l'immagine, indice della loro presenza. Sono state realizzate due distinte versioni di interfacce poiché, essendo il lavoro di tesi svolto in concomitanza con l'implementazione di *Social Mashlight*, si è deciso soltanto in un secondo momento di arricchire l'interazione dell'utente attraverso l'utilizzo di API esterne quali *Google Maps* e *Flickr* che potessero semplificare il processo di inserimento e visualizzazione delle informazioni rispettivamente utilizzate ed offerte dal servizio.

Definiamo di seguito gli strumenti presenti in ciascuna versione che consentono all'utente, data l'indicazione sull'area nella quale effettuare l'ispezione ed i servizi di elaborazione menzionati nella quinta sezione del quinto capitolo, di visualizzare all'interno di una mappa uno o più riscontri individuati dal sistema.

Drone form - Seconda scheda



The image shows a web application window titled "Drone Form". At the top, there are three tabs: "Sample / Monitor", "Search / Inspect" (which is selected and highlighted in blue), and "Track". Below the tabs, there is a form with the following fields and controls:

- Area**: A text input field containing the number "4".
- URL visita area**: A text input field containing the URL "http://localhost:65001".
- URL elaborazione dati**: A text input field containing the URL "http://localhost:65001".
- Ricerca**: A checkbox that is currently unchecked.
- Proseguì**: A button located below the "Ricerca" checkbox.

Figura 6.8: Sticker dedicato al controllo del APR per le sue funzionalità nella seconda tab

In figura 6.8 è mostrato il componente 'Drone Form', ma è stata selezionata la seconda tab etichettata con 'Search/Inspect', prima versione dell'interfaccia per l'inserimento dei dati necessari all'invocazione del servizio.

Al posto della griglia è presente un form in cui possono essere definiti i dati necessari corrispondenti alle definizioni presenti in tabella 5.2 a pagina 49:

- Il parametro ‘Area’ definisce i metri quadrati in cui il velivolo si muove.
- ‘URL visita area’ definisce il servizio che data un’area genera i punti da visitare al suo interno sotto forma di griglia.
- ‘URL elaborazione dati’ rinvia al servizio che elabora i risultati e risponde se un determinato punto è di interesse o meno.
- La checkbox ‘Search’ se spuntata indica al servizio che si vuole effettuare una Ricerca, se non selezionata viene effettuata una Ispezione che differisce dall’impiego alternativo come descritto nel capitolo terzo.

Drone form - Viewer

Il blocco utilizzato per la visualizzazione dei riscontri identificati dal servizio è il medesimo adottato nel servizio di Sample, mostrato in figura 6.7 a pagina 72.

Google Map

Questo form di inserimento fornisce al servizio dell’APR la seconda versione dell’interfaccia per l’inserimento dei seguenti dati necessari all’invocazione del servizio Search, corrispondenti anch’essi ai parametri descritti in tabella 5.2 a pagina 49:

- L’individuazione dell’area nella quale ricercare il pattern, selezionabile attraverso un’interfaccia grafica realizzata con l’utilizzo delle Google API relative al servizio Google Maps.
- ‘URL visita area’ definisce il servizio che data un’area genera i punti da visitare al suo interno sotto forma di griglia.
- ‘URL elaborazione dati’ rinvia al servizio che elabora i risultati e risponde se un determinato punto è di interesse o meno.
- La checkbox ‘Search’ se spuntata indica al servizio che si vuole effettuare una Ricerca, se non selezionato viene effettuata un’Ispezione

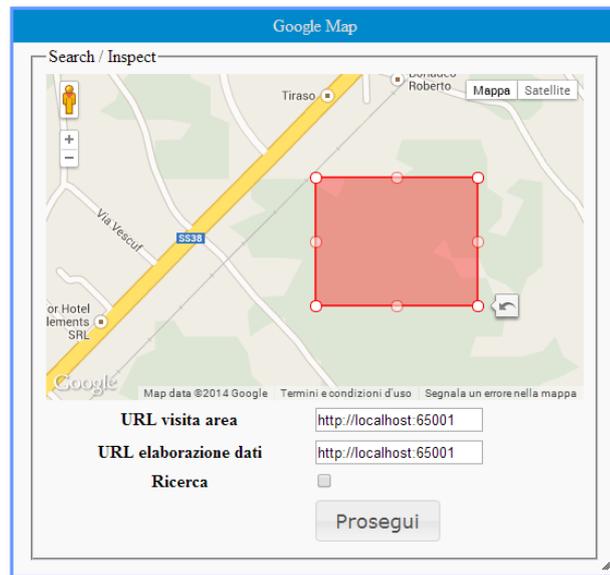


Figura 6.9: Selezione di un'estensione di area e parametri di ingresso per il servizio di Search.

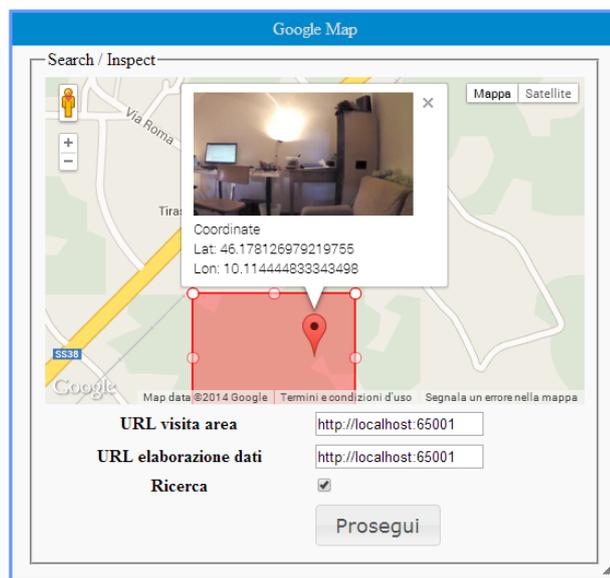


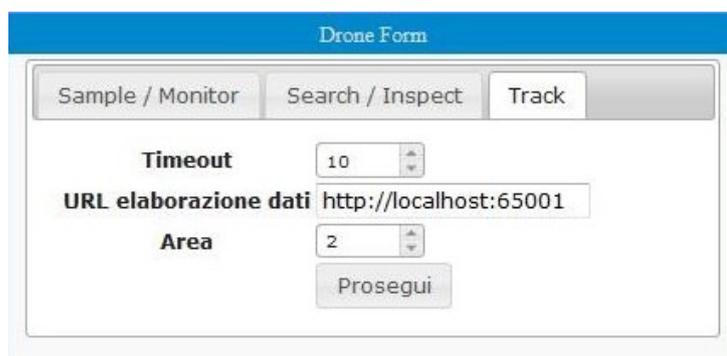
Figura 6.10: Punti di interesse visualizzati come Marker all'interno delle mappe di Google Maps.

Lo stesso Sticker è utilizzato al termine dell'esecuzione del servizio invocato per la visualizzazione delle coordinate geografiche relative ad un punto, selezionato in una cella appartenente allo Sticker 'Drone form - Viewer', all'interno della mappa di Google Maps con appositi *Marker* interattivi (Figura 6.10 nella pagina precedente).

6.2.3 Track

Definiamo gli strumenti necessari all'utente per poter avviare la ricerca di un target e visualizzarne il percorso ottenuto dall'inseguimento da parte dell'APR.

Drone form - Terza scheda



The image shows a software interface titled "Drone Form". It has three tabs: "Sample / Monitor", "Search / Inspect", and "Track". The "Track" tab is selected. Below the tabs, there are three input fields: "Timeout" with a value of 10, "URL elaborazione dati" with the value "http://localhost:65001", and "Area" with a value of 2. A "Prosegui" button is located below the "Area" field.

Figura 6.11: Sticker dedicato al controllo dell'APR per le sue funzionalità presenti nella terza tab

La terza scheda presente all'interno dello Sticker 'Drone Form' permette all'utente di interfacciarsi con il servizio 'Track'.

Prevede l'inserimento dei seguenti parametri corrispondenti a quelli indicati in tabella 5.3 a pagina 50 :

- Il timeout espresso in secondi, tempo massimo dopo il quale il servizio deve interrompere l'inseguimento dell'obiettivo.
- Il servizio Web relativo all'elaborazione delle immagini campionate dal velivolo;
- L'estensione massima dell'area, espressa in metri quadrati, entro la quale il velivolo può rimanere durante l'inseguimento; nel caso in cui l'oggetto inseguito fuoriesca dalla superficie indicata, il processo viene interrotto.

La chiamata al servizio comporta l'invocazione dello Sticker che mostra la griglia all'interno dello Sticker Drone form - Viewer, dove sono mostrati in tempo reale i punti nei quali il velivolo si è spostato, alla cui coordinata è avvenuta l'elaborazione dell'immagine precedentemente campionata.

Drone form - Viewer

Il blocco utilizzato per la visualizzazione dei riscontri identificati dal servizio è il medesimo adottato nel servizio di Sample, presente in figura 6.7 a pagina 72

6.3 Esperimenti di utilizzo

Sono di seguito presentati i due esperimenti di utilizzo relativi ai servizi Sample e Search realizzati all'interno del caso di studio. Per ciascuno sono evidenziati gli scenari che emulano, il relativo flusso di utilizzo che l'utente deve compiere per poter usufruire del servizio ed infine alcune note riguardanti le prestazioni di CPU e memoria annotate durante l'esecuzione degli applicativi server.

6.3.1 Scelta architetturale

Negli esperimenti di utilizzo che andremo ad approfondire non vi sarà la necessità di coordinare più velivoli né di archiviare una grande quantità di dati pertanto una macchina server interposta tra utente e velivolo non è indispensabile. Considerando un notebook quindi un apparato non minimale come device utilizzato dall'utente identifichiamo l'architettura a 2 livelli di deployment descritta nella seconda sezione del quarto capitolo come scelta di utilizzo. All'interno delle possibili configurazioni descritte adottiamo inoltre l'elaborazione dei servizi e l'elaborazione dei dati situate nel dispositivo di controllo la nostra scelta di utilizzo, poiché non vi è necessità né di elaborare dati nell'immediato all'interno del velivolo né di utilizzare un client minimale.

6.3.2 Drone Sample

L'esperimento seguente consente all'utente di usufruire del servizio Sample; dopo aver selezionato dei punti nei quali desidera delegare al velivolo il campionamento di immagini, ne visualizza il risultato.

Scenari emulati

Gli scenari emulati spaziano dalla perlustrazione di zone soggette a valanghe nella quale si è interessati a visualizzare determinati luoghi difficilmente raggiungibili da parte degli operatori di soccorso, alla mappatura di uno scavo archeologico nella quale l'automazione del processo di campionamento di immagini facilita la creazione della mappa del terreno, fino a considerare la ricostruzione digitale tridimensionale di un ghiacciaio in rapido scioglimento la quale necessita di immagini riprese in determinati punti difficili da raggiungere con altri mezzi di ripresa.

Flusso di utilizzo

1. Sulla griglia mostrata in figura 6.12, l'utente seleziona un gruppo di coordinate. I pulsanti selezionati si coloreranno di arancione e nella parte bassa dello Sticker verranno visualizzate le coordinate relative a partire dal punto di partenza centrale (contrassegnata dalla coppia $x=0$ e $y=0$);

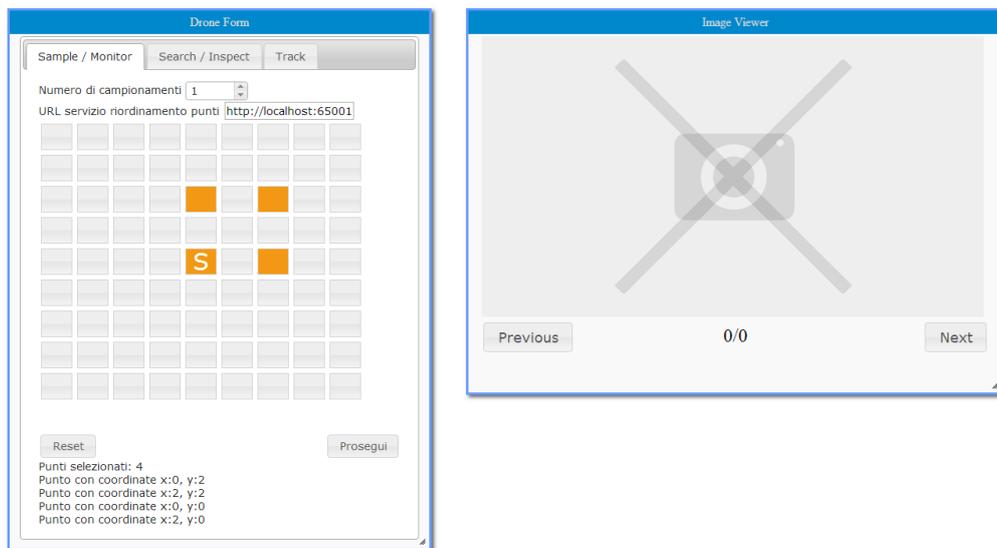


Figura 6.12: Selezione delle coordinate sulla griglia per poter indicare i punti nei quali il velivolo dovrà campionare le immagini.

2. Cliccando sul pulsante 'Prosegui', l'APR parte per l'attività di campionamento o monitoraggio, come descritte nella tassonomia presente nel secondo capitolo, in funzione al numero di campionamenti desiderati in ciascun punto e specificati nello spinner 'Numero di campionamenti'. Sulla coordinata dove è stato possibile effettuare la rilevazione, apparirà un'icona che rappresenta la presenza di una o più foto.
3. Cliccando sulla coordinata contrassegnata dall'icona di una immagine, il sistema visualizzerà come in figura 6.13 nella pagina seguente il set di immagini nel blocco 'Image Viewer'. Il numero di foto dipende dal parametro impostato nello spinner 'Numero di campionamenti'.

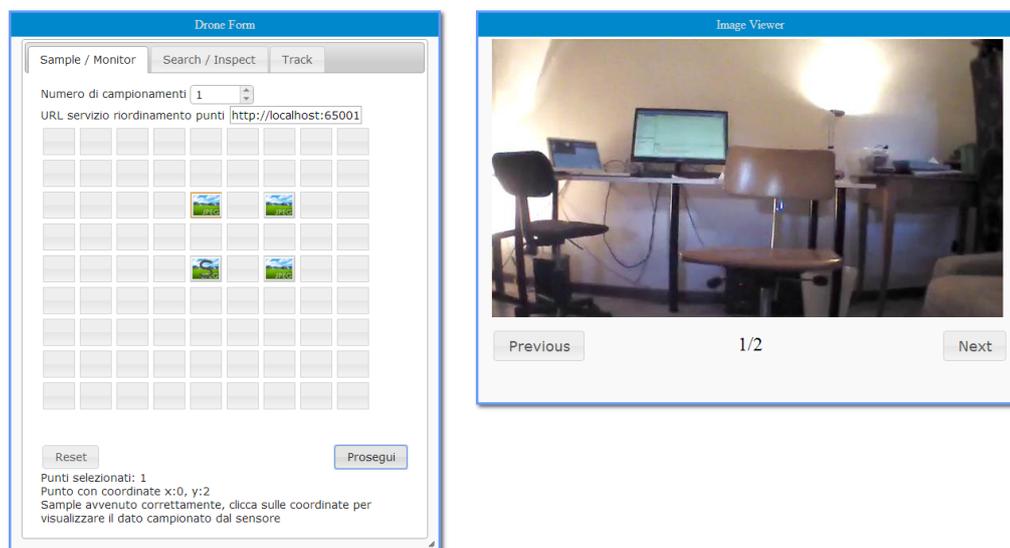


Figura 6.13: Visualizzazione delle immagini ottenute dal campionamento.

Analisi delle prestazioni

I dati prestazionali sono stati ottenuti attraverso l'utilizzo di un notebook Sony Vaio con processore Intel i5, 4GB di memoria RAM, Windows 7 SP1 come sistema operativo comprendente i soli processi essenziali al suo funzionamento e versione 7 aggiornamento 51 relativa alla Java Virtual Machine.

Attraverso l'utilizzo del tool 'Monitoraggio risorse' [14] presente in ambiente Windows, sono stati campionati i seguenti parametri durante ogni esecuzione di esperimento effettuata:

Utilizzo massimo memoria

Utilizzo di memoria in MB occupato dal processo comprendente l'esecuzione del server Web REST, la libreria di controllo relativa all'APR e la piattaforma di esecuzione del codice Java rappresentata dalla Java Virtual Machine. Gran parte della memoria utilizzata dal processo analizzato è da attribuire a quest'ultima.

Utilizzo medio CPU

Percentuale di utilizzo medio della CPU calcolata rispetto ad una media ottenuta sulla base dei dati registrati negli ultimi 60 secondi di esecuzione di ciascun esperimento.

Punti

Numero di punti nei quali si vuole campionare un'immagine con l'utilizzo dell'APR offerto dal sistema.

Campione	Utilizzo max memoria [MB]	Utilizzo medio CPU [%]	Punti
1	306	4,05	7
2	220	3,94	6
3	241	4,07	6
4	238	3,95	5
5	286	4,24	10
Media	258.2	4.05	6.8

Tabella 6.1: Campionamenti delle prestazioni del sistema relativi alle esecuzioni degli esperimenti del servizio Sample

Si rimanda alla quarta sezione dell'attuale capitolo l'analisi sui dati ottenuti, associati a quelli dell'esperimento successivo che ci apprestiamo a presentare.

6.3.3 Drone Search

L'esperimento consente all'utente di utilizzare il servizio Search.

E' stato realizzato in due versioni:

- La prima consente all'utente di specificare l'area di interesse nella quale eseguire l'ispezione di un target e visualizzarne in seguito i riscontri definiti dal punto e dalla relativa immagine.
- La seconda arricchisce la versione precedente integrando l'utilizzo della mappe fornite da Google Maps, per una più semplice definizione dell'area di interesse, ed il servizio di condivisione immagini Flickr per poter facilmente condividere i risultati ottenuti.

Scenari emulati

Gli scenari che meglio rappresentano questo esperimento di utilizzo sono molteplici e presenti in differenti contesti. Esempi citati nel secondo capitolo che

si adattano alle interfacce offerte sono il controllo dell'attraccaggio navi nel quale, impostati dei target si è interessati all'individuazione di imbarcazioni indesiderate, la perlustrazione di territori montani soggetti a valanghe con la quale si ricercano persone disperse, la perdita di olio in condutture marine identificabile dall'analisi di immagini fornite da un'alta quota e l'identificazione di rifiuti abusivi la quale può essere ottenuta da analisi cromatiche delle immagini campionate.

Flusso di utilizzo prima versione

1. Nel form mostrato nella parte sinistra della figura 6.14, l'utente definisce tutti i parametri per configurare il servizio REST.

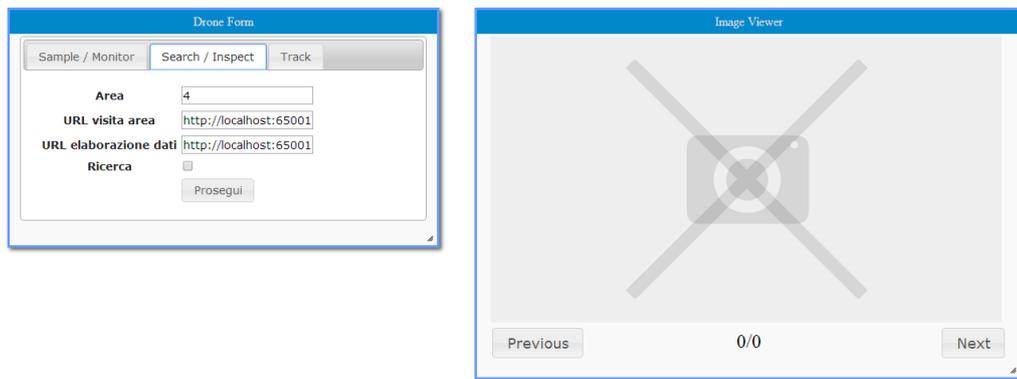


Figura 6.14: Definizione di tutti i parametri per avviare il servizio di Search.

2. Cliccando sul pulsante 'Prosegui', l'APR parte per l'attività di ricerca o ispezione descritta nella tassonomia presentata nel secondo capitolo, in funzione alla spunta della checkbox 'Ricerca'.

La griglia del Mashup verrà aggiornata come da figura 6.15 nella pagina seguente mostrando una nuova pagina con i due Stickers precedentemente descritti. Sulla coordinata dove è stato possibile effettuare la rilevazione, apparirà un'icona con una mappa che rappresenta che su quella coordinata sono presenti dei risultati. Affinchè i dati forniti siano consistenti, la posizione dell'APR relativa al suo decollo ed impostata per comodità al centro della griglia durante l'esecuzione dell'esperimento, implica che la posizione fisica del velivolo in tale utilizzo sia definita al centro dell'area di interesse considerata.

3. Cliccando sulla coordinata contrassegnata dall'icona, il sistema visualizzerà il set di immagini nel blocco 'Image Viewer' come da figura 6.15. Nell'immagine è presente una sola fotografia.

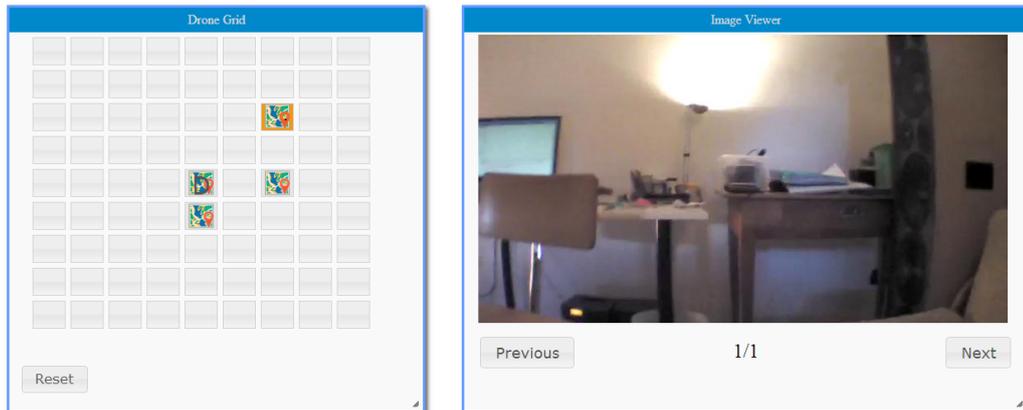


Figura 6.15: Visualizzazione delle immagini ottenute dall'invocazione del servizio Search.

Flusso di utilizzo seconda versione

Il processo durante l'interazione con l'utente, procede secondo le seguenti fasi:

1. Viene selezionata un'area rettangolare mediante l'utilizzo del componente Google Maps come illustrato in figura 6.16 nella pagina seguente che identifica la dimensione dell'area che il velivolo potrà sorvolare. L'APR dovrà essere posizionato nel suo centro affinché i dati siano consistenti con le coordinate fornite dal servizio di Google.

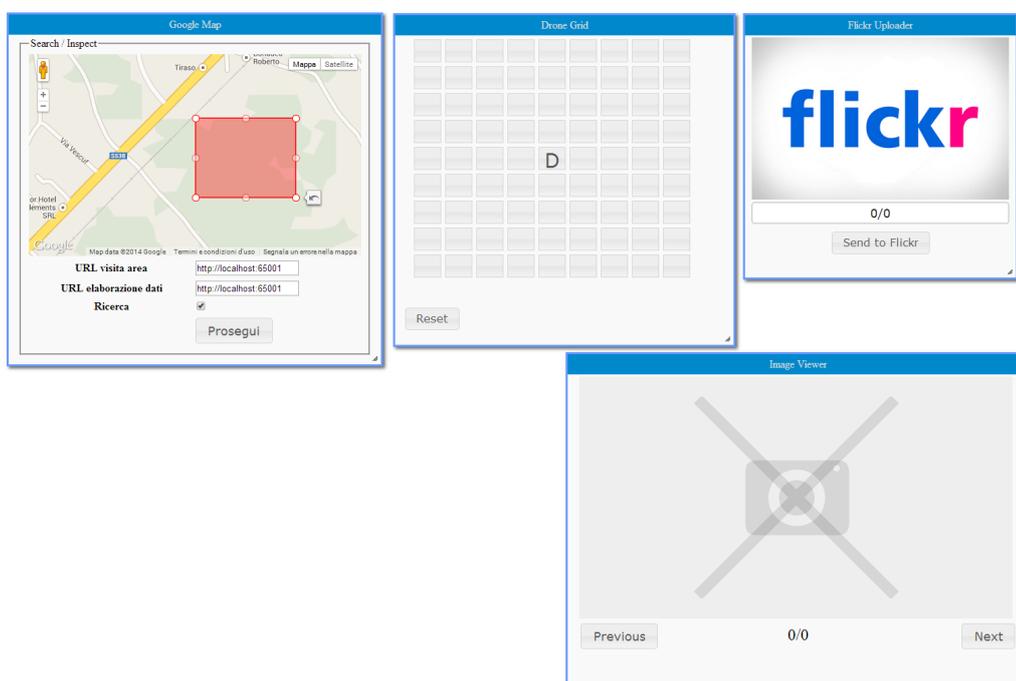


Figura 6.16: Definizione dell'ampiezza dell'area di esplorazione, all'interno di una mappa di Google Maps, corrispondente al parametro Area definito per il servizio di Search

2. Vengono forniti gli URL relativi ai servizi esterni, selezionando eventualmente la checkbox 'Ricerca' se interessati al relativo servizio.
3. Tramite la pressione sul pulsante 'Proseguì' si avvia l'attività dell'APR e l'elaborazione dei dati correlati.
4. Vengono visualizzati a schermo i riscontri positivi all'interno della griglia dei punti come da figura 6.17 nella pagina successiva.
5. Selezionando i punti contrassegnati da icone colorate, si ottiene un'immediata visualizzazione della coordinata all'interno della mappa di Google Maps e viene mostrata l'immagine campionata all'interno del visualizzatore di immagini come in figura 6.18 a pagina 86.
6. L'utente a questo punto può eventualmente effettuare il salvataggio remoto dei punti mediante l'utilizzo dello Sticker dedicato a Flickr attraverso la pressione del pulsante 'carica' come in figura 6.19 nella pagina successiva.

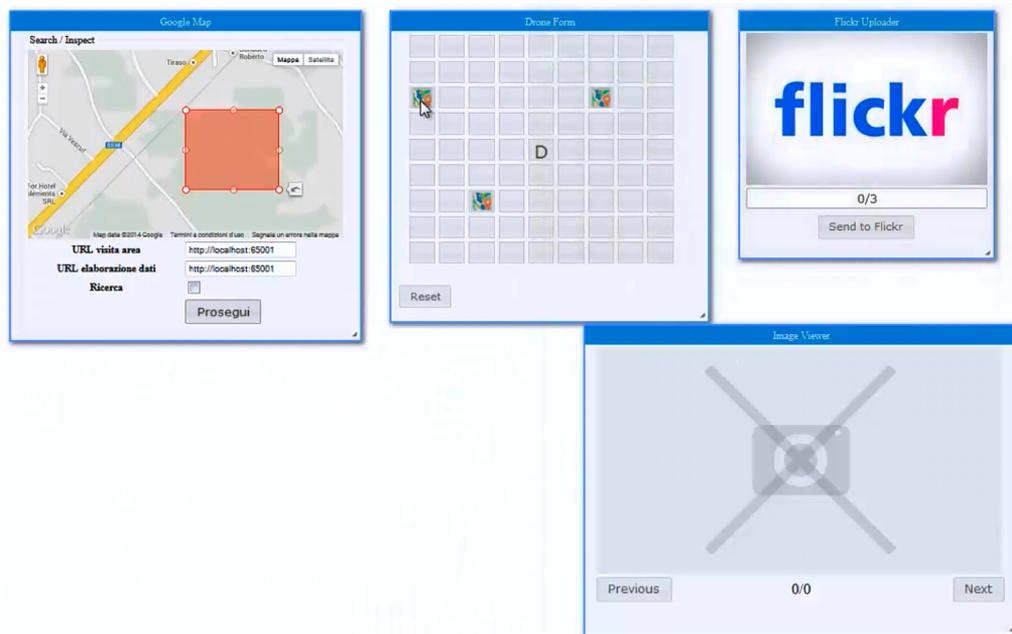


Figura 6.17: Punti interessati dai riscontri ottenuti dall'invocazione del servizio di Search, visualizzati all'interno della griglia dei punti



Figura 6.19: Caricamento in remoto delle rilevazioni ottenute attraverso l'utilizzo del servizio Search nel portale di condivisione immagini Flickr.



Figura 6.18: Visualizzazione approfondita dei risultati ottenuti dall'utilizzo del servizio di Search attraverso i marker presenti nella mappa di Google Maps rapportati all'elemento selezionato all'interno della griglia.

Analisi delle prestazioni ed utilizzo massimo della memoria

I dati prestazionali sono stati ottenuti con i medesimi strumenti hardware e software utilizzati durante l'analisi ottenuta nell'esperimento precedente. Resta invariata la descrizione dei parametri *Utilizzo massimo memoria* ed *utilizzo medio CPU* mentre si aggiunge un ulteriore parametro quale:

Numero riscontri

Numero dei riscontri positivi, programmati nel servizio di elaborazione dati presentato nella quinta sezione del capitolo quinto, ottenuti durante ciascuna esecuzione dell'esperimento.

Campione	Utilizzo max memoria [MB]	Utilizzo medio CPU	Num.Riscontri
1	310	3,86	4
2	323	4,56	6
3	338	3,81	3
4	329	3,97	4
5	333	4.15	5
Media	326.6	4.07	4.8

Tabella 6.2: Campionamenti delle prestazioni del sistema relativi alle esecuzioni degli esperimenti riguardanti il servizio Search

Viene ora presentato il sommario dei risultati ottenuti dall'esperimento considerato e da quello precedente.

6.4 Sommario dei risultati

Presentiamo ora le considerazioni, comuni alle prestazioni ottenute in entrambi gli esperimenti definiti, analizzando le due grandezze di interesse quali occupazione della memoria ed utilizzo della CPU.

Occupazione della memoria

Considerando 1,38 MB la dimensione del Bytecode, e 200 MB circa l'utilizzo di memoria RAM del processo in esecuzione i dati relativi al server REST *IceBreakRestServer* presentato nella terza sezione del quinto capitolo, alla libreria di supporto *YaDrone* adibita al controllo del velivolo presentata nella seconda sezione del quinto capitolo ed alla implementazione dei servizi descritta nella settima sezione del quinto capitolo, è necessario indirizzare l'alto consumo delle risorse relative alla memorizzazione alla piattaforma Java, nel dettaglio alla specifica implementazione della Java Virtual Machine fornita da Oracle presente nell'ambiente standard desktop utilizzato. Le misurazioni del consumo, nelle quali si è riscontrato un valore massimo di occupazione RAM del solo processo pari a 130 MB circa ottenuto come differenza del massimo valore di utilizzo del processo rispetto alla dimensione iniziale di esecuzione, possono essere pertanto considerate come upper-bound in quanto, identificando una piattaforma Java alternativa ed idonea ad essere eseguita in dispositivi con risorse hardware limitate, è possibile ridurre drasticamente l'utilizzo della memoria associata al processo e poter quindi considerare estremamente accettabile la massima occupazione di risorse necessaria.

Utilizzo della CPU

L'utilizzo della CPU da parte del processo ha influito in pochi punti percentuali sulle prestazioni offerte dal sistema, evidenziato il poco carico di lavoro necessario all'esposizione dei servizi Web da parte del sistema.

Capitolo 7

Conclusioni e Sviluppi futuri

Dopo aver presentato lo stato dell'arte riguardante aereomobili a pilotaggio remoto ed aver elencato i relativi casi d'utilizzo più comuni ai giorni nostri, abbiamo definito una tassonomia di impieghi suddivisi per funzionalità. Tale categorizzazione, analizzando gli elementi che contraddistinguono i requisiti funzionali di ciascun impiego, ci ha portati a definire le interfacce necessarie ad un loro utilizzo attraverso l'adozione di servizi Web come canale di interazione. Abbiamo in seguito considerato le possibili architetture idonee a supportare una piattaforma a servizi e le migliori tecnologie disponibili ai giorni nostri per poterle implementare. Sono stati poi forniti i dettagli relativi all'implementazione del codice, dal design programmatico ai diagrammi di sequenza che esplicano le relazioni presenti all'interno dei moduli software.

A dimostrazione del reale funzionamento dell'architettura, dopo avere descritto una delle molteplici possibilità per gestire l'interazione dell'utente con l'interfaccia del sistema, abbiamo infine proposto due esperimenti d'utilizzo completi che dimostrassero l'effettiva usabilità delle strutture Web esposte.

L'analisi dei risultati forniti dalle prestazioni ottenute durante gli esperimenti di utilizzo ha dimostrato, come già anticipato in diverse occasioni all'interno del lavoro di tesi e motivato dalla scelta di *IceBreakRestServer* definito nella terza sezione del capitolo quinto come implementazione di server REST minimale, che l'intero sistema di gestione dei servizi possa essere dislocato interamente a bordo di un velivolo come il *Parrot AR Drone 2.0*, descritto nella prima sezione del quinto capitolo. Il Bytecode dell'implementazione realizzata pari a soli 1,38 MB, composto dal server REST comprensivo della logica relativa ai servizi ed associato alla libreria di controllo dell'APR, ha presentato durante la sua esecuzione un'occupazione massima di memoria RAM pari a circa 130 MB, dimensione più che accettabile per uno spazio disponibile all'interno di un'architettura volatile moderna.

Tale codice, associato ad una Java Virtual Machine minimale ed ad un ridotto consumo della CPU caratterizzante il lavoro realizzato, rendono pertanto possibile una semplice integrazione del sistema proposto all'interno di una molteplicità di architetture ad oggi già esistenti, espandendo le loro possibilità di interazione con il mondo circostante ed aprendo nuove frontiere di acquisizione di dati da remoto.

Bibliografia

- [1] Aeroview international. <http://www.washingtonpost.com/wp-dyn/content/article/2005/11/27/AR2005112700739.html>.
- [2] Amazon prime air. <http://www.amazon.com/b?node=8037720011>.
- [3] Ar drone 2.0. <http://ardrone2.parrot.com>.
- [4] Ardrone p5. <http://kougaku-navi.net/ARDroneForP5/>.
- [5] Cranes and drones: Strange airfellows?
<http://www.fort.usgs.gov/RavenA/>.
- [6] Drones to watch over us highways.
<http://www.technewsdaily.com/16629-drones-watch-over-highways.html>.
- [7] Easy drone. <http://code.google.com/p/easydrone/>.
- [8] Extensible markup language (xml). <http://www.w3.org/XML/>.
- [9] Hargrave, aviation and aereomodelling - interdependent evolutions and histories. http://www.ctie.monash.edu/hargrave/rpav_home.html.
- [10] Icebreakrestserver. <https://code.google.com/p/icebreakrest/>.
- [11] Il telerilevamento aereo multispettrale per la mappatura degli incendi boschivi. www.terrasystem.it/articoli/MondoGIS_64.pdf.
- [12] Javadrone. <http://code.google.com/p/javadrone/>.
- [13] Mashups: The new breed of web app.
<http://www.ibm.com/developerworks/library/x-mashups/>.

- [14] Monitoraggio risorse. <http://windows.microsoft.com/it-it/windows7/open-resource-monitor>.
- [15] The pentagon's vision for the future of military drones. <http://motherboard.vice.com/blog/the-pentagons-vision-for-the-future-of-military-drones>.
- [16] Platform as a service (paas). <http://www.interoute.com/what-paas>.
- [17] Representational state transfer (rest). https://www.ics.uci.edu/~fielding/pubs/dissertation/rest_arch_style.htm.
- [18] Software as a service (saas). <http://www.saas.com/ta/hp.jsp>.
- [19] A suggested way to detect oil leakage. http://www.barnardmicrosystems.com/UAV/pipeline_monitoring/detect_oil_leakage.html.
- [20] The uninhabited aerial vehicle (uav) coffee project. <http://www.nasa.gov/missions/research/FS-2002-9-01-ARC.html>.
- [21] Yadrone. <http://vsi-wwww.informatik.uni-hamburg.de/oldServer/teaching/projects/yadrone/>.
- [22] M. Arjomandi. Classification of unmanned aerial vehicles.
- [23] Reg Austin. *Unmanned Air Systems*. April 2010.
- [24] Joseph Barnard. Unmanned aircraft in border patrol activities. Barnard Microsystems Limited.
- [25] C. H. Hugenholtz K. Whitehead, B. J. Moorman. Low-cost, on-demand aerial photogrammetry for glaciological measurement. Department of Geography, University of Calgary, Canada.
- [26] Arthur H. ter Hofstede Marlon Dumas, Wil M. van der Aalst. *Process-Aware Information Systems*. 2005.
- [27] Martinez-de Dios Anibal Ollero Merino, Caballero. Cooperative fire detection using unmanned aerial vehicles. Computer Vision and Intelligent Control Group University of Seville.

- [28] A. Puri. A survey of unmanned aerial vehicles (uav) for traffic surveillance.
- [29] Buschmann-Vorsmann Spiess, Bange. First application of the meteorological mini uav m2av. Institut für Luft- und Raumfahrtssysteme, Universität Braunschweig, Germany.
- [30] S. Wegener. Unmanned aerial vehicles, new frontiers in fire research and applications. Earth Science Division NASA Ames Research Center.
- [31] M. Weske W.M.P. van der Aalst, A.H.M. ter Hofstede. Business process management: A survey.