



POLITECNICO DI TORINO

Corso di Laurea in Ingegneria Informatica

Tesi di Laurea Magistrale

**Sviluppo di un sistema di
telemetria multiplatforma per il
veicolo autonomo**

Relatore

prof. Gianpiero Cabodi

Candidato

Luca GHIO

Supervisore Aziendale

ing. Giuseppe Giorgio

MARZO 2016

Sommario

Contesto

La presente tesi di laurea magistrale è stata realizzata in collaborazione con l'azienda Magneti Marelli, presso la sede di Venaria Reale (TO), sotto la guida dell'ingegnere Giuseppe Giorgio, ricercatore del gruppo ADAS Technologies, uno dei quattro gruppi di ricerca in cui è strutturato il gruppo CTO.

La tesi si situa nell'ambito della ricerca sul veicolo autonomo, la prossima rivoluzione nel campo dei trasporti a quattro ruote: un veicolo autonomo è un'autovettura in grado di guidare se stessa, senza l'intervento attivo di un operatore umano, grazie ai sistemi di controllo avanzato che interpretano le informazioni dai sensori per identificare i percorsi di navigazione appropriati, gli ostacoli e la segnaletica rilevante.

Il gruppo ADAS Technologies di Magneti Marelli sta sviluppando nuove tecnologie per la messa a punto di un proprio prototipo di veicolo autonomo, svolgendo dei test con simulazioni in varie piste d'Italia, come il Centro Sperimentale Balocco.

Obiettivi

Questa tesi ha lo scopo di presentare il progetto relativo allo sviluppo di un sistema di telemetria multiplatforma per il veicolo autonomo.

Il sistema di telemetria intende indirizzare un problema che attualmente affligge i ricercatori: il monitoraggio del veicolo durante i test è estremamente scomodo perché richiede l'accesso fisico al veicolo stesso da parte dei ricercatori.

L'applicazione è stata ideata e pensata dai ricercatori del gruppo ADAS Technologies per il controllo e il monitoraggio da remoto del loro veicolo autonomo, e verrà usata principalmente dai controllisti per osservare a distanza il veicolo in fase di test su pista.

La tesi, dopo una breve panoramica sullo stato dell'arte e sui prossimi sviluppi nell'ambito dei veicoli autonomi, si propone di descrivere l'architettura software del sistema di telemetria e le scelte progettuali che sono state affrontate nel corso del lavoro, confrontando tra loro soluzioni alternative e indicando i motivi per cui sono state scelte le soluzioni implementate.

Risultati ottenuti

Il sistema di telemetria sviluppato in questa tesi consiste in un server, in esecuzione all'interno del veicolo, che invia le informazioni sul veicolo tramite una connessione a Internet, e un client, in esecuzione sul dispositivo dell'utente, che riceve i dati dal server e li visualizza in tempo reale all'utente.

Il progetto è il risultato del lavoro congiunto di due tesisti: la componente server è stata sviluppata da Corrado Bellia, mentre la componente client è stata sviluppata da Luca Ghio.

Il server è composto da una parte embedded, sviluppata in C++, che legge i dati dalle reti interne del veicolo, e una parte di presentazione dati, sviluppata in Java, che permette ai client di collegarsi al server e invia loro i dati sul veicolo da visualizzare.

Il client è un'applicazione Web, sviluppata in HTML5, che permette all'utente di monitorare lo stato corrente del veicolo attraverso il proprio browser Web, con la

possibilità di visualizzare i grafici e di vedere la posizione del veicolo su una mappa.

Oltre al semplice monitoraggio dello stato veicolo, l'applicazione è in grado di registrare i grafici e di visualizzarli in un secondo momento in maniera sovrapposta ai grafici in tempo reale. Inoltre offre la possibilità di modificare le configurazioni dei moduli di controllo da remoto.

Un'ulteriore funzionalità è rappresentata dalla chat room, che permette ai ricercatori di comunicare con assoluta semplicità durante i test del veicolo.

Ringraziamenti

Si ringrazia il supervisore aziendale, l'ingegnere Giuseppe Giorgio, per la sua disponibilità durante l'andamento del progetto e i suoi consigli sulla stesura di questa tesi.

Indice

Sommario	II
Contesto	ii
Obiettivi	ii
Risultati ottenuti	iii
Elenco delle figure	XVI
Elenco delle tabelle	XVIII
1 Introduzione	1
1.1 Presentazione dell'azienda	1
1.1.1 Organizzazione aziendale	1
1.2 Sistemi ADAS	3
1.3 Il veicolo autonomo	5
1.3.1 Definizione	5
1.3.2 Caratteristiche	6
1.3.3 Livelli di automazione NHTSA	7
1.3.4 Livelli di automazione SAE	12
1.3.5 Effetti dell'introduzione dei veicoli autonomi	13
1.3.6 Legislazione	17
1.3.7 Prototipi di veicolo autonomo: tre casi di studio	18

1.3.8	Previsioni di commercializzazione	22
1.4	Sistemi di telemetria	24
1.4.1	Sistemi di telemetria per il campo automotive	25
2	Requisiti	29
2.1	Specifiche richieste da Magneti Marelli	29
2.1.1	Requisiti funzionali	30
2.1.2	Requisiti non funzionali	32
2.2	Scopo dell'applicazione	33
2.2.1	Monitoraggio dello stato di salute del veicolo	33
2.2.2	Visualizzazione dei grafici	34
2.2.3	Registrazione dei grafici	35
2.2.4	Configurazione dei moduli di controllo	35
3	Metodologia di progetto	37
3.1	Strumenti utilizzati	37
3.2	Documentazione	39
3.3	Tracciamento degli upstream	40
4	Panoramica dell'applicazione client	41
4.1	Che cos'è un'applicazione Web	41
4.1.1	Confronto con le applicazioni tradizionali	43
4.2	Pagine	43
4.2.1	Pagina iniziale	45
4.2.2	Grafici	46
4.2.3	Chat room	48
4.2.4	Telecamere	50
4.3	Pannelli	50
4.3.1	Mappa	50

4.4	Pagine di dialogo	53
4.4.1	Accedi	54
4.4.2	Modifica password	54
4.4.3	Riproduci registrazione remota	55
4.4.4	Riproduci registrazione locale	56
4.4.5	Scarica file di configurazione	57
4.4.6	Carica file di configurazione	57
4.4.7	Configura chat room	58
4.5	Dipendenze	59
5	Framework modulari usati	63
5.1	RequireJS	63
5.1.1	Motivi	64
5.1.2	AMD	64
5.1.3	Architettura dei moduli dell'applicazione	67
5.2	jQuery Mobile	68
5.2.1	Widget	68
5.2.2	Eventi di pagina	69
6	Scelte progettuali HTML5	73
6.1	WebSocket	73
6.1.1	Confronto con soluzioni alternative	74
6.2	Download	75
6.2.1	Salvataggio di file	76
6.2.2	Scaricamento in memoria RAM	77
6.2.3	Confronto con soluzioni alternative	77
6.3	Upload	79
6.3.1	Nome del file	79
6.3.2	Confronto con soluzioni alternative	80

7	Protocollo di comunicazione client-server	81
7.1	Formato di dati JSON	81
7.2	Endpoint del server	82
7.2.1	WebSocket principale	82
7.2.2	WebSocket di chat	83
7.2.3	Servlet di upload	83
7.3	Connessione di un client	83
7.3.1	Instaurazione di una connessione WebSocket	84
7.4	Sessione di registrazione	85
7.5	Sessione di riproduzione	86
7.6	Sessione di configurazione	88
7.7	Sessione di chat	90
8	Supporto dei browser	93
8.1	Accorgimenti per migliorare il supporto	94
8.1.1	Polyfill	94
8.1.2	Graceful degradation	94
8.1.3	Vendor prefix CSS	95
8.2	Analisi del supporto dei browser	96
8.2.1	Metodologia di analisi	96
8.2.2	Livelli di compatibilità	97
8.2.3	Risultati dell'analisi	99
9	Testing	101
9.1	Analisi statica del codice	101
9.1.1	Server	101
9.1.2	Client	102
9.2	Test con il simulatore	102
9.3	Utilizzo delle risorse	103

9.3.1	Utilizzo della larghezza di banda	103
9.3.2	Utilizzo della CPU	103
9.3.3	Utilizzo della memoria RAM	104
10	Conclusioni	105
10.1	Competenze acquisite	105
10.2	Possibili sviluppi futuri dell'applicazione	106
10.3	Casi d'uso alternativi dell'applicazione	107
A	Descrizione delle funzionalità HTML5 usate	109
A.1	Funzionalità richieste	109
A.1.1	Canvas	110
A.1.2	Elemento <code>meter</code>	110
A.1.3	Flexible Box Layout Module	110
A.1.4	Nuovi elementi semantici	111
A.1.5	Text API for Canvas	111
A.1.6	WebSocket	112
A.2	Funzionalità aggiuntive	112
A.2.1	2D Transforms CSS3	112
A.2.2	Attributo <code>download</code>	113
A.2.3	Colori CSS3	113
A.2.4	Elemento <code>video</code>	113
A.2.5	File API	113
A.2.6	Formato video MPEG-4/H.264	114
A.2.7	Funzionalità avanzate di <code>XMLHttpRequest</code> : tipo <code>blob</code>	114
A.2.8	Gestione della cronologia di sessione	114
A.2.9	Media Query CSS3	115
A.3	Abbellimenti facoltativi	115
A.3.1	API <code>CSS.supports()</code>	115

A.3.2	Attributo <code>accept</code>	115
A.3.3	Attributo <code>autocomplete</code>	116
A.3.4	Attributo <code>spellcheck</code>	116
A.3.5	<code>calc()</code> come valore di unità CSS	116
A.3.6	Constraint validation API	117
A.3.7	CSS Font Loading	117
A.3.8	CSS Hyphenation	117
A.3.9	Cursori CSS3	118
A.3.10	ECMAScript 5 Strict Mode	118
A.3.11	Elemento <code>progress</code>	118
A.3.12	Favicon PNG	118
A.3.13	Funzionalità avanzate di <code>XMLHttpRequest</code> : tipo <code>json</code>	119
A.3.14	Geolocalizzazione	119
A.3.15	Internationalization API	119
A.3.16	Ligature	119
A.3.17	Menu contestuale	120
A.3.18	Proprietà CSS <code>will-change</code>	120
A.3.19	Proprietà CSS3 <code>border-radius</code>	120
A.3.20	Proprietà personalizzate CSS	121
A.3.21	Selettori CSS3	121
A.3.22	Valore CSS <code>unset</code>	122
A.3.23	Valore <code>none</code> della proprietà CSS <code>user-select</code>	122
A.3.24	Web font <code>@font-face</code>	122
A.3.25	Web Notifications	122
B	Formato dei messaggi JSON	125
B.1	Messaggi del WebSocket principale	126
B.1.1	Informazioni sul veicolo	126

B.1.2	Account utente	128
B.1.3	Registrazioni	130
B.1.4	File di configurazione	131
B.1.5	Stato delle sessioni	132
B.1.6	Errori generali	132
B.2	Messaggi del WebSocket di chat	133
B.2.1	Accessi alla chat room	133
B.2.2	Messaggi di chat	134
B.3	Messaggi della servlet di upload	135
C	Documentazione tecnica dei moduli RequireJS	137
C.1	Modulo main	137
C.2	Moduli delle pagine	137
C.2.1	Modulo page-cameras	138
C.2.2	Modulo page-chat	138
C.2.3	Modulo page-graph	138
C.2.4	Modulo page-home	138
C.3	Moduli delle pagine di dialogo	138
C.3.1	Modulo dialog-changepassword	138
C.3.2	Modulo dialog-configurechat	139
C.3.3	Modulo dialog-downloadconfigurationfile	139
C.3.4	Modulo dialog-localrecording	139
C.3.5	Modulo dialog-login	139
C.3.6	Modulo dialog-remoterecording	139
C.3.7	Modulo dialog-uploadconfigurationfile	139
C.4	Moduli dei pannelli	140
C.4.1	Modulo panel-map	140
C.5	Moduli del core	140

C.5.1	Modulo <code>accelerationbars</code>	140
C.5.2	Modulo <code>configuration</code>	140
C.5.3	Modulo <code>gauge</code>	141
C.5.4	Modulo <code>gauge-speedometer</code>	141
C.5.5	Modulo <code>gauge-tachometer</code>	141
C.5.6	Modulo <code>graph</code>	142
C.5.7	Modulo <code>graph-latacceleration</code>	142
C.5.8	Modulo <code>graph-longacceleration</code>	143
C.5.9	Modulo <code>graph-speed</code>	144
C.5.10	Modulo <code>header</code>	145
C.5.11	Modulo <code>jsonsubscriber</code>	145
C.5.12	Modulo <code>login</code>	146
C.5.13	Modulo <code>map</code>	147
C.5.14	Modulo <code>recording</code>	147
C.5.15	Modulo <code>statusfooter</code>	148
C.5.16	Modulo <code>steeringwheel</code>	149
C.5.17	Modulo <code>weather</code>	149
C.5.18	Modulo <code>websocket</code>	149
C.5.19	Modulo <code>websocket-chat</code>	150
C.5.20	Modulo <code>websocket-server</code>	151
C.6	Moduli di utilità	152
C.6.1	Modulo <code>utils</code>	152
C.6.2	Modulo <code>utils-array</code>	153
C.6.3	Modulo <code>utils-loader</code>	154
C.6.4	Modulo <code>utils-notifications</code>	154
C.7	Moduli di plug-in	155
C.7.1	Modulo <code>jquery.mobile-pageevents</code>	155
C.7.2	Modulo <code>jquery.mobile-widthresponsivepanel</code>	156

Elenco delle figure

1.1	Organigramma aziendale.	2
1.2	Il prototipo del veicolo di Magneti Marelli.	19
1.3	Il team di VisLab che nel 1998 fu protagonista nel pionieristico progetto ARGO.	20
1.4	Il prototipo della Google Car.	21
1.5	L'evoluzione dei sistemi ADAS.	23
1.6	Le proiezioni, ottimistiche e pessimistiche, sul numero dei veicoli autonomi venduti e sul numero totale dei veicoli autonomi in circolazione, secondo l'analisi del Victoria Transport Policy Institute.	24
1.7	Ferrary Telemetry: esempio di app mobile per la telemetria.	26
2.1	Diagramma di contesto completo per l'applicazione.	29
2.2	Mock-up dell'interfaccia per l'applicazione di telemetria richiesta da Magneti Marelli.	30
2.3	Rappresentazione esemplificativa dell'interfaccia grafica del software HMI attualmente in uso.	34
4.1	Pagina iniziale.	44
4.2	Pagina "Grafici".	47
4.3	Pagina "Chat room".	49
4.4	Pagina "Telecamere".	51

4.5	Pannello “Mappa”	52
4.6	Pagina di dialogo “Accedi”	54
4.7	Pagina di dialogo “Modifica password”	55
4.8	Pagina di dialogo “Riproduci registrazione remota”	55
4.9	Pagina di dialogo “Riproduci registrazione locale”	56
4.10	Pagina di dialogo “Scarica file di configurazione”	57
4.11	Pagina di dialogo “Carica file di configurazione”	58
4.12	Pagina di dialogo “Configura chat room”	59
5.1	Grafo delle dipendenze tra i moduli RequireJS.	66
5.2	Diagramma degli eventi di pagina jQuery Mobile (modello a pagina singola).	70
7.1	Diagramma di sequenza della connessione di un client, con l’instaurazione della connessione WebSocket al server principale.	84
7.2	Diagramma di sequenza di una sessione di registrazione.	86
7.3	Diagramma di sequenza di una sessione di riproduzione.	87
7.4	Diagramma di sequenza di una sessione di configurazione.	88
7.5	Diagramma di sequenza di una sessione di chat.	90
A.1	Il layout delle barre di accelerazione nella pagina iniziale è pesantemente basato sui flexible box.	110

Elenco delle tabelle

1.1	Riepilogo dei livelli di automazione di guida SAE per i veicoli su strada.	11
8.1	Tabella di riepilogo del supporto delle funzionalità HTML5 da parte dei browser Web.	98