

DeSpeect

PRESENTED BY Graphite

M. Focchiatti · S. Modena

M. Rizzo · G. Rossetti · K. Silvestri M.

Smaniotto · C. Tessarolo



DeSpeect



Scopo del progetto

fornire un'interfaccia grafica utilizzabile come strumento di supporto all'utilizzo di plugin sulla piattaforma Speect



L'utente avrà la possibilità di salvare i grafi generati a schermo dall'applicazione



Il funzionamento dell'applicazione sarà garantito su un sistema Linux Ubuntu versione 16.04 o superiore

Tecnologie

Abbiamo diviso le tecnologie in due gruppi



Tecnologie di sviluppo

- ✓ Speect v1.1.0-69-g65f4
- ✓ QT v5.9 LTS
- ✓ CMAKE v3.10.2
- ✓ Ubuntu v16.04.3 LTS
- ✓ Travis CI



Tecnologie di supporto

- ✓ Google Drive
- ✓ Hangouts
- ✓ Slack
- ✓ Wrike
- ✓ LaTeX
- ✓ Git

Tecnologie di sviluppo

Speed v1.1.0-69-g65f4

QT v5.9 LTS

CMAKE v3.10.2

Ubuntu v16.04.3 LTS

Travis CI



Tecnologie concorrenziali

- OpenMary
- Idlak



Aspetti negativi

- Scarsa leggibilità del codice
- Necessita di non banali procedure di configurazione



Cosa ci ha spinto a sceglierla

- Tecnologia vincolata dalla Proponente



Tecnologie concorrenziali

- GTK+
- wxWidgets



Aspetti negativi

- Performance più basse in relazione di alcune tecnologie
- Installazione richiede molto spazio



Cosa ci ha spinto a sceglierla

- Semplicità d'uso
- Ampio utilizzo in ambito aziendale
- Buona stabilità



Tecnologie concorrenziali

- GNU Makefile
- Qmake



Aspetti negativi

- Molte funzionalità dipendono dalla versione specifica di CMAKE
- La sintassi è disuniforme e confusionaria
- La documentazione introduttiva è scarsa e presenta pochi esempi



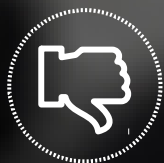
Cosa ci ha spinto a sceglierla

- Tecnologia parzialmente vincolata dalla Proponente
- Tecnologia già utilizzata da Speect



Tecnologie concorrenziali

- Microsoft Windows
- Apple MacOS



Aspetti negativi

- Ubuntu non supporta (o non supporta completamente) alcuni software di utilizzo comune o selezionati dal gruppo per problemi organizzativi



Cosa ci ha spinto a sceglierla

- La Proponente richiede garanzia di funzionamento del prodotto su questo specifico sistema operativo

Travis CI



Tecnologie concorrenziali

- Circleci
- Wercker



Aspetti negativi

- Necessita di software di terze parti per personalizzazioni avanzate



Cosa ci ha spinto a sceglierla

- Semplicità d'uso
- Perfetta integrazione con lo strumento di versionamento Github e la tecnologia CMAKE

Tecnologie di supporto

Google Drive

Hangouts

Slack

Wrike

LaTeX

Git

Tecnologie di supporto

Google drive



Tecnologie concorrentziali

- Dropbox
- Mega



Cosa ci ha spinto a sceglierla

- Ottima integrazione con l'ecosistema Google

Tecnologie di supporto

Hangouts



Tecnologie concorrentziali

- Skype



Cosa ci ha spinto a sceglierla

- Ottima integrazione con l'ecosistema Google

Tecnologie di supporto

➞ Slack



Tecnologie concorrentziali

- Azendoo
- eXo Platform



Cosa ci ha spinto a sceglierla

- Ottima integrazione con molte delle altre tecnologie usate dal gruppo e compatibilità con i maggiori sistemi operativi

Tecnologie di supporto

➞ Wrike



Tecnologie concorrentziali

- Asana
- Bitrix24



Cosa ci ha spinto a sceglierla

- Ottima compatibilità con le maggiori piattaforme

Tecnologie di supporto

LaTeX



Tecnologie concorrenti

- Markdown
- Microsoft Word



Cosa ci ha spinto a sceglierla

- Ottima qualità tipografica dei documenti e chiara distinzione tra contenuto e formattazione

Tecnologie di supporto

Git



Tecnologie concorrenti

- Subversion



Cosa ci ha spinto a sceglierla

- Semplicità e minimo ingombro di risorse
- Buona integrazione con le altre tecnologie



PoC

Introduce le tecnologie,
i framework e le librerie selezionate
per lo sviluppo del prodotto e ne dimostra adeguatezza e
grado di integrazione in relazione agli obiettivi di progetto.

Consiste di un'interfaccia grafica
realizzata tramite librerie Qt che, ricevuta una stringa in input,
invoca un metodo della libreria Speect
per estrarre la relation token e la stampa sotto forma di grafo
insieme all'input normalizzato in lettere minuscole.

La build del PoC sfrutta la tecnologia CMAKE
per la corretta compilazione di Speect.

Problemi rilevati

Problemi risolti e funzionalità sviluppate

Compilazione e configurazione di Speect tramite CMAKE

Interfaccia grafica realizzata con Qt

Stampa del grafo (parzialmente manipolabile) relativo a una relation dell'utterance di Speect

Portabilità di Speect in C++

Parziale implementazione del software per il testing automatico

Problemi irrisolti

Importazione e esportazione del grafo

Completa manipolazione del grafo

Completa configurazione del software per il testing automatico

Problemi risolti e funzionalità sviluppate



Compilazione in C++ di QT e Speect

Non è stato semplice compilare il nostro sorgente includendo le due librerie perché in Speect manca parte del supporto necessario alla compilazione tramite cmake.

Oltre che compilare cmake, riusciamo ad utilizzare il progetto in QtCreator.

Configurazione di Speect

Abbiamo notato vari problemi di installazione della libreria causati da un link esterno non stabile, a quel punto abbiamo modificato il file di configurazione correggendo il problema.

Problemi risolti e funzionalità sviluppate



Disegno e manipolazione di elementi grafici attraverso il cursore

Da una prima analisi pensavamo che questa parte fosse difficile ma non abbiamo riscontrato grossi problemi nel realizzarla.

Incapsulamento di Speect tramite oggetti

Non si sono presentati problemi.

Parziale implementazione del software per il testing automatico

Viene notificato su slack un eventuale errore di compilazione.

Problemi risolti e funzionalità sviluppate



Dai problemi risolti abbiamo scoperto che

La trasformazione del grafo HRG in audio tramite i plugin forniti dal proponente producono un audio distorto se non eseguiti come prima operazione

graphite.swe@gmail.com