Verbale Esterno

Informazioni Documento

Data approvazione	15 dicembre 2017
Responsabile	Matteo Rizzo
Redattore	Manfredi Smaniotto
Verificatore	Giulio Rossetti
Distribuzione	Prof. Tullio Vardanega
	Prof. Riccardo Cardin
	Gruppo Graphite
Uso	Esterno
Recapito	graphite.swe@gmail.com

Indice

1 Informazioni generali

1.1 Informazioni sull'incontro

- **Luogo:** Aula P300, Complesso Paolotti (Videochiamata);
- **Data:** 15-12-2017;
- **Orari** 14:00 - 14:40;
- **Partecipanti del gruppo:** Marco Focchiatti, Samuele Modena, Matteo Rizzo, Giulio Rossetti, Kevin Silvestri, Manfredi Smaniotto, Cristiano Tessarolo;
- **Partecipanti esterni:** Dr. Giulio Paci.

1.2 Ragioni dell'incontro

Chiarimenti su dubbi emersi riguardo i requisiti del prodotto richiesto nel capitolato d'appalto proposto da MIVOQ SRL ([C3]). Tale capitolato è reperibile al seguente link:

http://www.math.unipd.it/~tullio/IS-1/2017/Progetto/C3.pdf

2 Resoconto

Nel corso dell'incontro sono stati chiariti i dubbi sollevati dal gruppo sul prodotto da realizzare per il capitolato proposto da MIVOQ SRL, nello specifico sono stati delucidati i seguenti punti:

- Illustrata interfaccia grafica dimostrativa inclusa nel capitolato d'appalto, con spiegazione del significato e delle funzionalità richieste dagli elementi che la compongono;
- Chiarimento sul funzionamento generale della tecnologia *Speect*$_{\mathrm{G}}$ di cui farà largo uso il progetto;
- Chiarimento sul livello di manipolazione richiesto dei nodi del *grafo*$_{\mathrm{G}}$ offerto in output all'utente;
- Illustrazione della struttura dei file *json*$_{\mathrm{G}}$ di cui fa uso Speect.

Seguono le principali domande poste durante l'incontro e la sintesi delle relative risposte ricevute:

1. **Lo scopo del progetto è di realizzare un'interfaccia grafica per Speect oppure un debugger per lo stesso?**
   Il capitolato richiede la realizzazione di un'interfaccia grafica per Speect che stampi su schermo i grafi relativi ai vari stati interni dello stesso per una specifica esecuzione. Tali grafi verranno utilizzati per semplificare la ricerca di errori in applicazioni che utilizzano Speect ed in questo senso il prodotto da realizzare svolgerà una funzione di debugging.

2. **In che modo l'interfaccia da realizzare dovrà interagire con Speect?**
   L'interfaccia da realizzare invocherà i metodi forniti da Speect sfruttandoli per la stampa dei grafi.

3. **Dall'interfaccia dimostrativa presente nel capitolato, si evince che l'applicazione dovrà poter ricevere in input file .json ed elaborarli. Quale ruolo hanno questi file all'interno di Speect? La loro struttura ci verrà fornita o starà a noi progettarla?**
   Speect sfrutta la tecnologia JSON per interagire con dati persistenti (per esempio, a scopo di inizializzazione). La struttura dei file con cui interagisce è già definita.

4. **A che livello deve essere possibile per l'utente interagire con il grafo stampato dal prodotto da realizzare?**
   L'utente deve poter modificare i nodi, importare ed esportare i grafi.

5. **Salvare un grafo comporta il salvataggio di una immagine o lo stato del grafo visibile in quel momento in modo da ricaricarlo rapidamente sul programma?**
   Con "salvare un grafo" si intende salvare uno stato del grafo capace anche di far riprendere la compilazione da un istante preciso in avanti. Speect dovrebbe avere già dei metodi appositi per poter effettuare l'esportazione dei grafi con l'unica condizione che ogni elemento del grafo deve essere serializzabile.

6. **Il software che deve essere prodotto deve essere un interfaccia interna a Speect oppure un programma a parte?**
   Deve essere prodotto un programma a sè stante che inizializzi Speect al suo interno.

7. **Come funziona l'interfaccia grafica e qual'è la funzione di ogni elemento mostrato sull'immagine esempio che è possibile trovare sul capitolato?**
   La colonna sinistra mostra 3 liste di cui la prima in alto sono gli utterance processor elencati all'interno del file.json compilato; essi sono i primi elementi che all'atto della compilazione vengono elencati.
   Le ulteriori liste presenti nella parte sottostante servirebbero per modificare il file .json con dei feature processor e degli utterance processor che dovrebbe esser possibile comporre all'interno del file.json da salvare successivamente.
   Tale parte è comunque opzionale al fine del funzionamento dell'applicazione.
   La parte sinistra deve poter permettere di eseguire singolarmente ogni utterance processor presente nel file.json interagendo con i bottoni (quelli che nella figura sono dei quadratini erano intesi come dei cerchi).
   I campi di testo presenti nella parte alta dell'interfaccia d'esempio contengono l'indirizzo di:

   - il file json da cui si sono caricati i vari utterance processor nel path denominato voice;
   - il path in cui è puntato un nodo quando un utente, dopo aver selezionato un nodo, interagisce con l'interfaccia grafica.

   L'esigenza di sapere quale nodo è puntato in un dato path è data dal fatto che utilizzando più plugin Speect opera su una grande quantità di directory e non è sempre chiaro in quale di esse il programma sta lavorando. Per evitare di seguire il percorso a mano si vuole rendere il processo di ricerca automatico.

8. **Quando viene premuto uno dei bottoni relativi agli utterance processor presenti nella colonna sinistra dell'interfaccia grafica vengono eseguiti tutti gli utterance processor precedenti oppure si esegue solo quello selezionato?**
   Viene eseguito solo l'utterance processor selezionato. Il grafo mostrato sarà comunque quello cumulativo. Potrebbe succedere che se il feature processor non trova gli elementi di cui ha bisogno venga lanciato un errore.
   Devono essere invece eseguite tutte le utterance processor di un certo tipo nel caso in cui venga premuto in alto a destra il relativo tasto di avvio di un certo tipo di utterance.
   Possibilmente si deve poter creare in partenza una utterance senza un

tipo, ma non è detto che al momento si possa fare.
Devono quindi essere percorribili sia la strada in cui un utente desidera creare il grafo in modo completo sia selezionando le singole utterance dal menù a sinistra.

9. **Deve essere mostrata la compilazione del file voice.json step-by-step oppure è sufficiente mostrare il risultato finale?**
   L'importante è che venga mostrato il risultato finale, come già avviene con Speect test.

10. **Cosa deve permettere di fare il programma nel caso in cui mancassero delle sequenze specifiche per avviare correttamente la costruzione del grafo?**
    Nulla di più di segnalare un errore, poichè sarà il programmatore che usa l'interfaccia a modificare il file.json per il corretto funzionamento del proprio plugin.
    Requisito opzionale potrebbe essere di rendere editabili le utterance type all'interno dell'interfaccia (per evitare il ricaricamento del file.json).

11. **Il caricamento del file.json deve avvenire in contemporanea alla costruzione del grafo oppure devono essere 2 azioni distinte?**
    Sono 2 azioni diverse che avvengono in 2 momenti diversi.

12. **Il file.json aperto durante la costruzione del grafo viene aperto in automatico durante il caricamento di un salvataggio del grafo?**
    Non è certo che il file.json venga salvato assieme al file di salvataggio del grafo, quindi dovrà essere verificata più avanti la possibilità di caricare entrambe le cose allo stesso tempo.
    Non bisogna aspettarsi che Speect abbia già le funzionalità utili per il nostro scopo; dovrebbe però permettere di crearle in modo abbastanza semplice.

13. **Tra inizializzazione di Speect tramite funzione apposita e l'utilizzo di Speect nella utterance ci sono dei metodi di cui non sappiamo la funzione; è possibile sapere cosa fanno di preciso?**
    Servono a caricare dei plugin esterni a Speect per caricare dei formati supportati da Speect a runtime. Essi vengono usati normalmente per salvare un qualsiasi elemento generico di Speect in formato EBML.

3 Tracciamento delle decisioni

- **VE-2017-15-12.1:** Integrazione del contenuto della risposta alla domanda 4 all'interno dei casi d'uso del documento *Analisi dei Requisiti v1.0.0*.