

saudio-sim Report

Alessandro Sciarrillo, Niccolò Mussoni, Alex Presepi, Simone Lugaresi

March 2, 2022

Contents

1	Analisi	2
1.1	Requisiti	2
1.2	Analisi e modello del dominio	3

Chapter 1

Analisi

Il software ha l'obiettivo di permettere all'utente di sperimentare un ascolto realistico in relazione alla posizione all'interno di un ambiente personalizzabile, attraverso la scelta di diversi luoghi di ascolto e tipi di speakers, permettendogli di spostare in tempo reale sia le sorgenti sonore che il punto d'ascolto da cui si ascolta.

1.1 Requisiti

Nell'analisi dei *requisiti* dell'applicazione si dovrà spiegare cosa l'applicazione dovrà fare. Non ci si deve concentrare sui particolari problemi, ma esclusivamente su cosa si desidera che l'applicazione faccia. È consigliato descrivere separatamente i requisiti funzionali (quelli che descrivono l'effettivo comportamento dell'applicazione) da quelli non funzionali (requisiti che non riguardano direttamente aspetti comportamentali, come sicurezza, performance, eccetera).

Elementi positivi

- Si fornisce una descrizione in linguaggio naturale di ciò che il software dovrà fare.
- Gli obiettivi sono spiegati con chiarezza, per punti.
- Se il software è stato commissionato o è destinato ad un utente o compagnia specifici, il committente viene nominato.
- Se vi sono termini il cui significato non è immediatamente intuibile, essi vengono spiegati.
- Vengono descritti separatamente requisiti funzionali e non funzionali.
- Considerato a un paio di pagine un limite ragionevole alla lunghezza della parte sui requisiti, in quello spazio si deve cercare di chiarire *tutti* gli

aspetti dell'applicazione, non lasciando decisioni che impattano la parte “esterna” alla discussione del design (che dovrebbe solo occuparsi della parte “interna”).

Elementi negativi

- Si forniscono indicazioni circa le soluzioni che si vogliono adottare
- Si forniscono dettagli di tipo tecnico o implementativo (parlando di classi, linguaggi di programmazione, librerie, eccetera)

Esempio

Il software, commissionato dal gestore del centro di ricerca “Aperture Laboratories Inc.”, mira alla costruzione di una intelligenza artificiale di nome GLaDOS (Genetic Lifeform and Disk Operating System). Per intelligenza artificiale si intende un software in grado di assumere decisioni complesse in maniera semi autonoma sugli argomenti di sua competenza, a partire dai vincoli e dagli obiettivi datigli dall'utente.

Requisiti funzionali

- La suddetta intelligenza artificiale dovrà occuparsi di coordinare le attività all'interno delle camere di test di Aperture, guidando l'utente attraverso un certo numero di sfide di difficoltà crescente. Una camera di test è un ambiente realizzato da Aperture Laboratories Inc. al fine di mettere alla prova le proprie tecnologie di manipolazione dell'ambiente. All'interno della camera di test, un soggetto qualificato è incaricato di sfruttare gli strumenti messi a disposizione da Aperture per risolvere alcuni rompicapi. I rompicapi sono di tipo fisico (ad esempio, manipolazione di oggetti, pressione di pulsanti, azionamento di leve), e si ritengono conclusi una volta che il soggetto riesce a trovare l'uscita dalla camera di test.
- Il piano preciso ed il numero delle sfide sarà variabile, e GLaDOS dovrà essere in grado di adattarsi dinamicamente e di fornire indicazioni di guida.
- La personalità di GLaDOS dovrà essere modificabile.
- GLaDOS dovrà essere in grado di comunicare col reparto cucina di Aperture, per ordinare torte da donare agli utenti che completassero l'ultima camera di test con successo.

Requisiti non funzionali

- GLaDOS dovrà essere estremamente efficiente nell'uso delle risorse. Le specifiche tecniche parlano della possibilità di funzionare su dispositivi alimentati da una batteria a patata.

1.2 Analisi e modello del dominio

In questa sezione si descrive il modello del *dominio applicativo*, descrivendo le *entità* in gioco ed i rapporti fra loro. Si possono sollevare eventuali aspetti particolarmente impegnativi, descrivendo perché lo sono, senza inserire idee circa possibili soluzioni, ovvero sull'organizzazione interna del software. Infatti, la fase di analisi va effettuata **prima** del progetto: né il progetto né il software esistono nel momento in cui si effettua l'analisi. La discussione di aspetti propri del software (ossia, della *soluzione* al problema e non del problema stesso) appartengono alla sfera della progettazione, e vanno discussi successivamente.

È obbligatorio fornire uno schema UML del dominio, che diventerà anche lo scheletro della parte “entity” del modello dell'applicazione, ovvero degli elementi costitutivi del modello (in ottica MVC - Model View Controller): se l'analisi è ben fatta, dovrete ottenere una gerarchia di concetti che rappresentano le entità che compongono il problema da risolvere. Un'analisi ben svolta **prima** di cimentarsi con lo sviluppo rappresenta un notevole aiuto per le fasi successive: è sufficiente descrivere a parole il dominio, quindi estrarre i sostantivi utilizzati, capire il loro ruolo all'interno del problema, le relazioni che intercorrono fra loro, e reificarli in interfacce.

Elementi positivi

- Viene descritto accuratamente il modello del dominio.
- Alcuni problemi, se non risolvibili in assoluto o nel monte ore, vengono dichiarati come problemi che non saranno risolti o saranno risolti in futuro.
- Si modella il dominio in forma di UML, descrivendolo appropriatamente.

Elementi negativi

- Manca una descrizione a parole del modello del dominio.
- Manca una descrizione UML delle entità del dominio e delle relazioni che intercorrono fra loro.
- Vengono elencate soluzioni ai problemi, invece della descrizione degli stessi.
- Vengono presentati elementi di design, o peggio aspetti implementativi.
- Viene mostrato uno schema UML che include elementi implementativi o non utili alla descrizione del dominio, ma volti alla soluzione (non devono vedersi, ad esempio, campi o metodi privati, o cose che non siano equivalenti ad interfacce).

Esempio

GLaDOS dovrà essere in grado di accedere ad un'insieme di camere di test. Tale insieme di camere prende il nome di percorso. Ciascuna camera è composta di challenge successivi. GLaDOS è responsabile di associare a ciascun challenge un insieme di consigli (suggestions) destinati all'utente (subject), dipendenti da possibili eventi. GLaDOS dovrà poter comunicare coi locali cucina per approntare le torte. Le torte potranno essere dolci, oppure semplici promesse di dolci che verranno disattese.

Gli elementi costitutivi il problema sono sintetizzati in

La difficoltà primaria sarà quella di riuscire a correlare lo stato corrente dell'utente e gli eventi in modo tale da generare i corretti suggerimenti. Questo richiederà di mettere in campo appropriate strategie di intelligenza artificiale.

Data la complessità di elaborare consigli via AI senza intervento umano, la prima versione del software fornita prevederà una serie di consigli forniti dall'utente.

Il requisito non funzionale riguardante il consumo energetico richiederà studi specifici sulle performance di GLaDOS che non potranno essere effettuati all'interno del monte ore previsto: tale feature sarà oggetto di futuri lavori.