

Alma Mater Studiorum · Università di Bologna

SCUOLA DI SCIENZE
Corso di Laurea in Informatica Magistrale

**Riabilitazione cognitiva
delle funzioni esecutive
nella Sclerosi Multipla:
un approccio basato sul planning**

Relatore:
Chiar.mo Prof.
Mauro Gaspari

Presentata da:
Daniele Baschieri

Correlatore:
Dott.
Floriano Zini

Sessione 2
Anno Accademico 2016/2017

Sommario

Più di 2.5 milioni di persone nel mondo soffrono di sclerosi multipla, una malattia fortemente debilitante che ne mina capacità motorie e cognitive. In Italia la percentuale è piuttosto alta con 57.000 malati, ovvero 1 italiano su 1000. In ambito cognitivo una rilevante percentuale di pazienti (40-70%) sviluppa deficit su differenti domini cognitivi, attenzione, memoria e funzioni esecutive. Queste ultime sono le più complesse con rilevante impatto nella vita di tutti i giorni. Un paziente con importanti deficit nelle funzioni esecutive riscontra in generale gravi conseguenze sulla qualità della vita. Allo stato attuale non esistono cure per trattare la patologia ma esistono efficaci sistemi di riabilitazione cognitiva per contenere e limitare la progressione della malattia almeno per quanto riguarda l'ambito psicologico. I pazienti sono soliti effettuare sessioni riabilitative in ospedali specializzati per il trattamento della patologia e queste sessioni richiedono personale dedicato tra cui psicologi, logopedisti e altri operatori sanitari. Le sessioni riabilitative possono essere individuali o di gruppo fino ad una decina di partecipanti assistiti da uno o più operatori. Gli esercizi proposti nella maggior parte dei casi vengono svolti con carta e penna o con giocattoli che aiutino il paziente a fraternizzare con l'esercizio. Durante lo svolgimento degli esercizi i pazienti raccontano agli operatori le loro scelte e descrivono le loro azioni per risolverlo correttamente.

In questi anni, fin dal 1993 è iniziato un processo di digitalizzazione dell'esperienza riabilitativa, molti esercizi cartacei sono stati tradotti in programmi per computer con un significativo aumento della partecipazione da parte dei pazienti e di soddisfazione nell'utilizzo. Non esistendo fino ad ora programmi specifici per trattare la sclerosi multipla il gruppo di ricerca del DISI del prof. Mauro Gaspari ha progettato e realizzato in collaborazione con il dott. Sergio Stecchi, ex direttore del UOSI Riabilitazione sclerosi multipla dell'ospedale Bellaria di Bologna, e il centro sclerosi multipla dell'ospedale di Fidenza-Vaio (PR), un software chiamato MS-Rehab con il chiaro intento di fornire alle strutture sanitarie un programma completo e semplice per la riabilitazione cognitiva della sclerosi multipla. Il software si presenta come una applicazione web, corredata di tutti gli strumenti utili all'operatore per gestire il processo di riabilitazione dei pazienti, fornendo strumenti per la gestione di riabilitazione dei singoli pazienti o di gruppi. L'operatore può visualizzare le statistiche delle performance raggiunte dai pazienti, uno storico sugli esercizi svolti e sulla situazione sanitaria e riabilitativa del paziente. Per il paziente invece il programma mette a disposizione un insieme di esercizi nei tre ambiti di attenzione, memoria e funzioni esecutive.

L'obiettivo di questa tesi è la realizzazione di un esercizio atto alla riabilitazione delle funzioni esecutive. Esercizi di questo tipo richiedono competenze di pianificazione soprattutto se viene richiesto di affrontare problemi simili a quelli che i pazienti possono incontrare nella vita reale. Alcuni esempi sono presenti in software per la riabilitazione cognitiva ma hanno molte limitazioni e affrontano un numero limitato di scenari. L'approccio proposto in questo lavoro è invece più ambizioso, si vuole realizzare un programma capace di generare innumerevoli esercizi e scenari per offrire al paziente sempre nuove sfide riabilitative. Così facendo si potrà coinvolgerlo in un crescendo di difficoltà in modo da stimolare sempre più la sua capacità critica di elaborazione delle informazioni e le sue funzioni esecutive.

In questa tesi si descrive quindi la progettazione e lo sviluppo di un programma in grado di risolvere problemi di pianificazione di una giornata basata su pianificazione automatica e per utilizzarlo è stata progettata un'interfaccia grafica minimale che permette al paziente di compiere le azioni essenziali per risolvere il problema, l'interazione tra le azioni del paziente e le previsioni del planner permettono di capire se la strategia del paziente è adeguata e se il paziente ha commesso errori. Si è anche scritto un generatore di problemi al fine di generare un numero pressoché illimitato di nuovi scenari.

Nel capitolo 1 verrà descritta la Sclerosi Multipla e il programma MS-Rehab, nel capitolo 2 verrà descritto con maggior dettaglio il dominio cognitivo delle funzioni esecutive, nel capitolo 3 vi sarà un'introduzione del linguaggio usato per la pianificazione e nel capitolo 4 il pianificatore vero e proprio. Il capitolo 5 è invece la descrizione dell'interfaccia e del software sviluppato. Infine i capitoli 6 e 7 sono per test di usabilità e conclusioni.

Indice

1	Background	3
1.1	Sclerosi multipla	3
1.2	Stato dell'arte nella riabilitazione cognitiva	5
1.2.1	La riabilitazione cognitiva	5
1.2.2	Domini cognitivi	7
1.2.3	Sistemi per la riabilitazione cognitiva	8
1.2.4	Limiti dei sistemi esistenti	10
1.3	Il sistema MS-Reahb	11
1.3.1	Sezione operatori	12
1.3.2	Sezione per il paziente	14
1.3.3	Esercizi disponibili	14
1.3.4	Realizzazione	14
2	Funzioni esecutive	17
2.1	Esempi di funzioni esecutive	17
2.2	Riabilitazione delle funzioni esecutive	18
2.3	Riabilitazione ecologica	19
2.4	Esempi di esercizi riabilitativi	20
2.5	Realizzazioni attuali	21
3	Planning e PDDL	23
3.1	Pianificazione	23
3.2	PDDL	24
3.2.1	Modellare con PDDL	25
4	Il pianificatore	28
4.1	PDDL applicato al processo risolutivo	28
4.2	Mantenere generale il problema	33
4.3	Il generatore	36

5	La progettazione dell'applicazione	38
5.1	Casi d'uso	38
5.2	Diagramma di sequenza	41
5.3	Strutturazione degli oggetti	42
5.4	Base di Dati di MS-Rehab	43
5.5	Grafica	45
5.6	Descrizione del compito	46
5.7	Valutazione delle performance	49
6	Test	51
6.1	Efficienza dei generatori	51
6.2	Test Usabilità	53
6.2.1	Protocollo di user testing	53
6.2.2	Risultati del test sugli utenti	55
6.2.3	Perfezionamenti successi alla valutazione	59
7	Conclusioni	62
7.1	Conclusioni	62
7.2	Sviluppi futuri	63

Capitolo 1

Background

Per comprendere meglio il lavoro svolto è necessario fare alcune doverose premesse riguardanti la sclerosi multipla, il suo relativo decorso e i sintomi che essa suscita nei pazienti. Occorrerà anche studiare lo stato dell'arte della riabilitazione cognitiva della sclerosi multipla e di come questa si stia rapidamente trasformando in questi ultimi anni. Infine sarà necessario introdurre il sistema chiamato MS-Rehab e le sue funzionalità descrivendo nel dettaglio le premesse alla base del programma e le sue potenzialità nella riabilitazione cognitiva.

1.1 Sclerosi multipla

La sclerosi multipla (SM) è una malattia autoimmune cronica demielinizzante che affligge il sistema nervoso centrale (ovvero il cervello e midollo spinale, causando diversi sintomi in uno spettro ampio). È una malattia che interessa più di 400.000 persone nel Nord America, circa 500.000 in Europa e 2.5 milioni di individui nel mondo. L'Italia è tra i paesi con il maggior numero di casi: si stima un totale di 57.000 malati, circa un italiano su 1000. La Sardegna è forse la regione più colpita con una media di 1 abitante su 700 a contrarre la malattia. La sclerosi multipla colpisce prevalentemente individui di sesso femminile, con un rapporto di circa 3 a 1 rispetto ai maschi. La malattia esordisce prevalentemente negli adulti sulla trentina, ma si sono verificati casi di bambini affetti da questa patologia [10]. Le cause scatenanti sono ad oggi ancora sconosciute, sebbene è probabile che vi sia una combinazione di fattori genetici, ambientali e infettivi. Sebbene la sclerosi multipla non sia considerata una malattia ereditaria, vi sono variazioni genetiche che hanno messo in luce la capacità di aumentare il rischio di sviluppo della malattia [12]. Tra i fattori ambientali figurano sia quelli di origine infettiva che quelli di natura non infettiva. La malattia è più comune tra persone che vivono lontane

dall'equatore, con una correlazione maggiore relativa alla poca esposizione solare. L'infanzia gioca un ruolo fondamentale nello sviluppo della malattia, una ricerca correlata con il periodo di migrazione da luoghi equatoriali mostra che i soggetti che migrano prima dei 15 anni hanno lo stesso rischio di persone nate lontane dall'equatore mentre coloro che migrano dopo i 15 anni di età mantengono il rischio del loro paese d'origine. Fumo, tossine e solventi possono inoltre inficiare nello sviluppo della patologia [3][19]. Tra le cause infettive che possono rappresentare fattori scatenanti per la malattia è l'ipotesi che queste possano essere determinate dalla presenza di un qualche microrganismo, tra questi il più probabile è il virus di Epstein-Bar che se contratto in giovane età attiverrebbe i processi che in età adulta portano alla patologia vera e propria.

La sclerosi multipla è caratterizzata dalla progressiva degenerazione della mielina, una sostanza grassa che riveste le fibre nervose, gli assoni, questi filamenti partono dai neuroni e trasmettono i segnali in uscita. Gli assoni svolgono quindi il ruolo di output dei neuroni e di input per altri neuroni, la loro forma ricorda la "chioma di un albero" con una radice priva di diramazioni nella parte prossimale al neurone, ricca e folta in quella distale. Nella sclerosi multipla, per cause non ben chiarite, si scatena una reazione autoimmune in cui i linfociti T, attaccano la guaina mielinica che riveste gli assoni. Questo comporta la comparsa di aree demielinizzate, chiamate placche che rallentano o interrompono la propagazione del segnale negli assoni. Se in un individuo sano la conduzione degli impulsi nervosi è di 100 m/s, un individuo affetto da sclerosi multipla può vedere ridotta la velocità di trasmissione di più del 90% con una velocità di propagazione di 5 m/s. La causa è da imputare proprio alla assenza della guaina mielinica. I linfociti T sono cellule del sistema immunitario coinvolte nella risposta immunitaria cellulo-mediata ovvero una specifica componente atta a colpire le cellule che presentano anomalie quali danneggiamenti sulla superficie causati da virus, o cellule infettate da batteri intracellulari e cellule tumorali che presentano antigeni tumorali. La malattia è definita autoimmune poiché cellule sane della mielina vengono erroneamente scambiate per malate e attaccate dai linfociti. Le ferite provocate dalla rottura della guaina mielinica sono chiamate sclerosi, da lì il nome della malattia. Questa si presenta non in una regione circoscritta del sistema nervoso centrale ma su tutta la sua superficie e per questo è chiamata multipla. Tenuto conto che la malattia può colpire qualsiasi punto del sistema nervoso centrale senza alcuna correlazione tra le aree coinvolte, il quadro clinico dei malati può variare in modo significativo. Anche lo sviluppo della malattia non è lineare ma presenta periodi di maggiore intensità e periodi di parziale ripresa con un decorso difficile da prevedere. La sclerosi multipla si manifesta con un quadro clinico molto variegato tra i quali figurano tipicamente i seguenti sintomi: fatica, disturbi dell'attività e

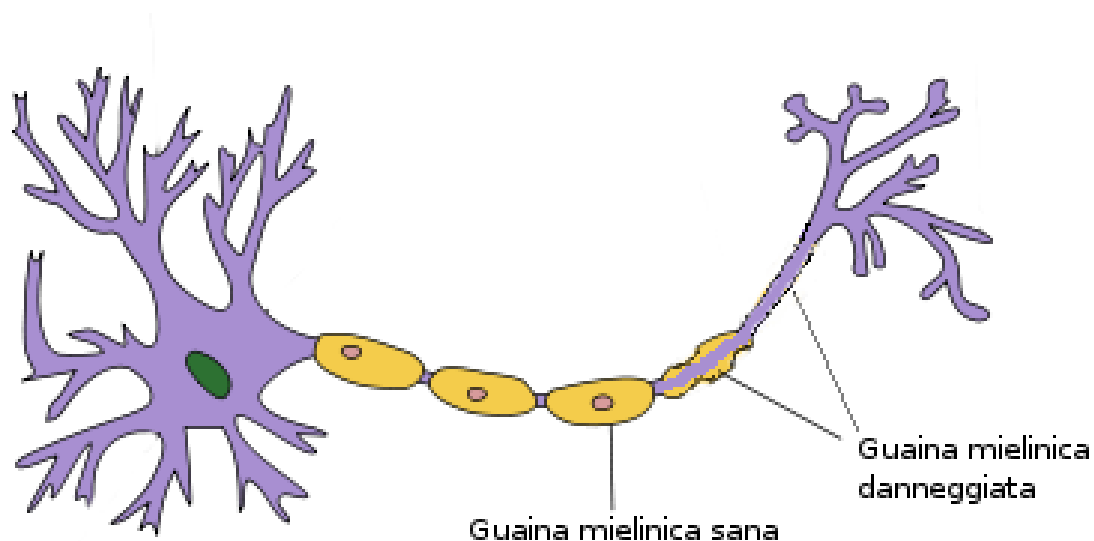


Figura 1.1: Esempio di assone affetto da sclerosi multipla

della coordinazione motoria, parestesie, disturbi della vista, disturbi della parola, disturbi vescicali e/o intestinali, disturbi dell'equilibrio, disturbi sessuali, dolore, depressione, disturbi cognitivi.

La presenza di deficit cognitivi nei soggetti affetti da sclerosi multipla è un fatto noto fin dal 1800, data della scoperta della malattia, ma soltanto negli ultimi 20-30 anni si è iniziato uno studio sistematico dei disturbi cognitivi che ne derivano. Si calcola che almeno tra il 40% e il 70% dei pazienti affetti da sclerosi multipla abbia disfunzioni cognitive di una certa entità [4]. Come è evidente dalla figura 1.2 le aree colpite dalla sclerosi multipla sono piuttosto variegate, con una particolare propensione verso la memoria e la velocità nello sviluppo dei processi.

1.2 Stato dell'arte nella riabilitazione cognitiva

1.2.1 La riabilitazione cognitiva

La sclerosi multipla è una malattia che non è facilmente inquadrabile dal punto di vista dei disturbi cognitivi che determina, pertanto risulta necessario uno studio preventivo e una pianificazione individuale del processo di riabilitazione per

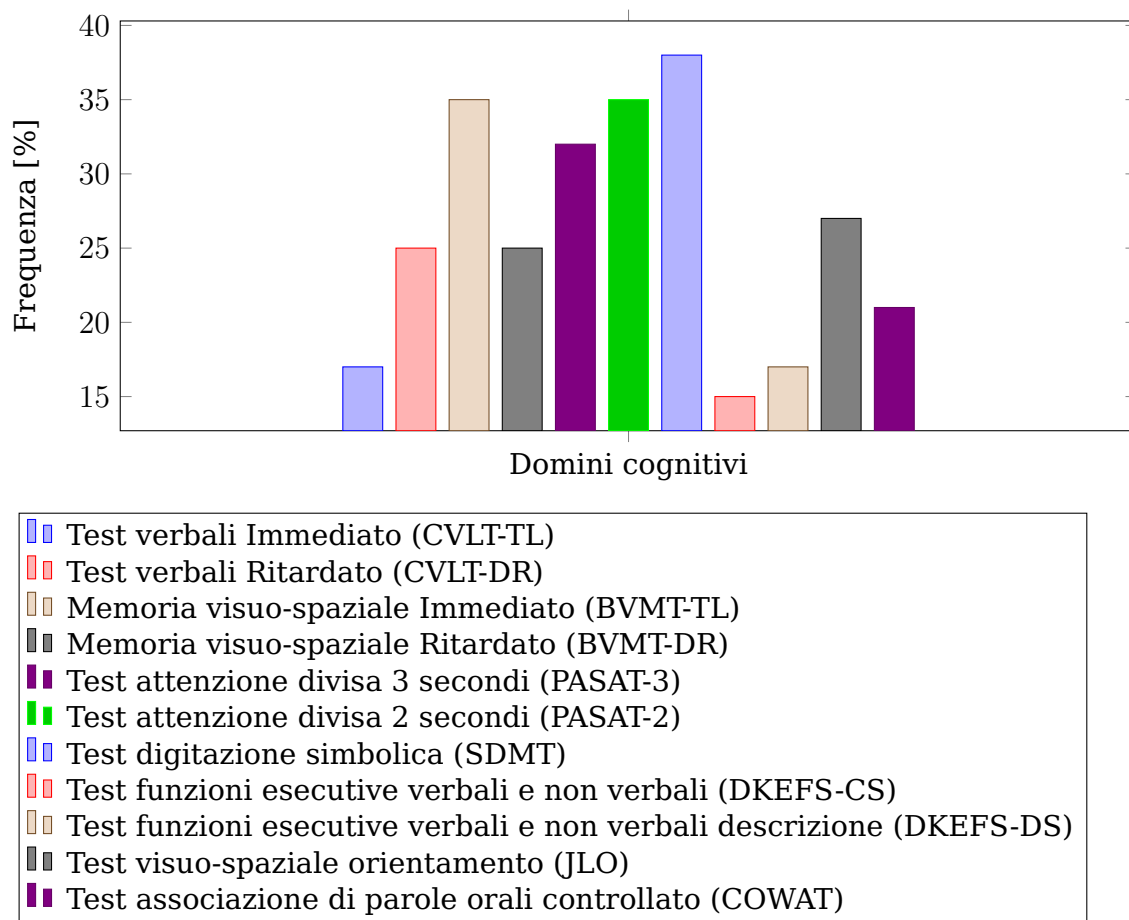


Figura 1.2: Frequenza dei pazienti affetti da menomazioni [4].

ciascun paziente. La principale difficoltà dei pazienti con sclerosi multipla è individuabile nella codifica delle informazioni più che nel loro successivo recupero dalla memoria. Occorre perciò individuare strategie riabilitative che operino in questa direzione ad esempio, uno studio del dottor Chiaravalotti et al. [9] mette in luce i seguenti punti chiave:

- La ripetizione potrebbe non aiutare a migliorare le rievocazione, ma una migliore codifica dell'informazione sicuramente ha un impatto positivo nella rievocazione.
- A causa del rallentamento nella velocità delle funzioni cognitive occorre fornire un tempo adeguato per l'acquisizione dell'informazione.

- È importante diminuire al minimo gli stimoli distrattori durante la fase di codifica.

Nello stesso studio viene anche proposto un meccanismo di codifica chiamato *generation effect*, si suppone infatti che le informazioni codificate dallo stesso paziente siano più semplici da recuperare. Pertanto sono da prediligere tutti quegli esercizi che forzano il paziente a produrre una codifica propria dell'informazione. In questa ottica gli esercizi di completamento di parole o di frasi possono rappresentare un buon punto di partenza.

1.2.2 Domini cognitivi

Attenzione

L'attenzione è quella capacità cognitiva che permette agli individui di focalizzare l'attività mentale selezionando stimoli prefissati nell'ambiente, sia esterno che interno. L'attenzione può essere selettiva o dinamica. Si tratta di un'abilità cognitiva di primaria importanza in quanto permette, in pazienti sani, di sfruttare correttamente le risorse cognitive per l'analisi o la elaborazione di specifici stimoli percepiti attraverso i sensi. L'attenzione è fondamentale nell'apprendimento in quanto un deficit nell'attenzione rende estremamente difficile memorizzare e ricordare cose importanti. Bambini affetti da deficit di attenzione hanno spesso difficoltà a seguire istruzioni, elaborare informazioni, ascoltare, difficoltà a concentrarsi su una cosa specifica [20].

Un modello clinico proposto da Sohlberg e Mateer [22] propone una divisione schematica dell'attenzione in varie tipologie:

Attenzione focalizzata: Rappresenta la capacità di rispondere a uno specifico stimolo di tipo visivo, uditivo o tattile.

Attenzione sostenuta: È la capacità di mantenere un buon livello di risposta agli stimoli mentre si sta svolgendo un compito differente o comunque continuativo nel tempo.

Attenzione selettiva: La capacità di mantenere la concentrazione su un particolare stimolo, o un aspetto dello stimolo, mentre si sopprimono volontariamente le risposte ad altri stimoli distrattori.

Attenzione alternata: Indica l'abilità di direzionare il focus attentivo allo svolgimento di compiti che richiedono risorse cognitive differenti come senti-

re un suono o guardare immagini. Denota una certa flessibilità mentale al cambiamento degli stimoli.

Attenzione divisa: Al contrario della attenzione alternata che richiede un passaggio tra differenti stimoli l'attenzione divisa richiede di far fronte contemporaneamente a più stimoli tra loro differenti.

Memoria

La memoria è quella funzionalità cognitiva che consente la codifica, l'immagazzinamento e il recupero delle informazioni anche a distanza di tempo. La memoria riguarda molteplici ambiti cognitivi distinti, l'attenzione, ad esempio, gioca un ruolo fondamentale nella prima fase di percezione e successiva codifica dell'informazione. La codifica è complessa e impatta sulla qualità dei ricordi e sulla loro durata nel tempo. Si può perciò suddividere il processo della memoria in funzione di quanto debba essere la permanenza dell'informazione memorizzata si parla quindi di: memoria a lungo termine e memoria a breve termine. La memoria a breve termine è deputata al mantenimento di informazioni per brevi periodi di tempo e in un numero limitato. La capacità massima di informazione immagazzinabile nella memoria a breve termine prende il nome di *span*, ed è una caratteristica che varia da persona a persona. Tipicamente il numero massimo di elementi si aggira tra i 3 e i 5 fino a casi di 7 elementi. La memoria a lungo termine invece ha capacità di immagazzinamento illimitato e per periodi lunghi, potenzialmente per sempre.

Funzioni Esecutive

Le funzioni esecutive si riferiscono a quelle abilità coinvolte nella pianificazione, sequenziamento e organizzazione del comportamento. Dato che sarà oggetto di questa tesi una più approfondita analisi delle funzioni esecutive sarà poi presentata nel prossimo capitolo.

1.2.3 Sistemi per la riabilitazione cognitiva

Sono stati tentati diversi approcci alla riabilitazione cognitiva, in primo luogo vi è la riabilitazione tradizionale in ospedale ad opera di logopedisti e psicologi, gli esercizi sono sottoposti in forma cartacea al paziente, l'operatore segue lo svolgimento dell'esercizio saltuariamente intervenendo per variarne alcune caratteristiche o per chiedere dettagli sulle scelte del paziente. L'approccio è diretto e

richiede personale specializzato dedicato al singolo paziente o a gruppi di pazienti. Sebbene questa strategia sia da preferirsi per alcuni aspetti poiché offre agli operatori il massimo controllo sul paziente e sul suo stato riabilitativo, presenta alcune immediate problematicità: seguendo questo metodo risulta molto difficile realizzare interventi continuativi, il paziente deve attendere le sessioni di riabilitazione alle volte anche settimanali o mensili, il costo di questo tipo di riabilitazione è decisamente elevato poiché richiede la presenza stabile di personale dedicato.

Alcune vie più moderne sono state già percorse come l'utilizzo di programmi riabilitativi computerizzati, il paziente in ambito ospedaliero effettua la sessione riabilitativa sfruttando strumenti quali un computer con alcune periferiche dedicate. La riabilitazione, in questo caso maggiormente interattiva, è sempre guidata dall'operatore, ma il sistema può suggerire avanzamenti di difficoltà automatici tra un esercizio e l'altro. Si è visto che l'approccio computerizzato alla riabilitazione cognitiva è un valido strumento a disposizione degli operatori, e può seriamente aiutare i pazienti nel processo riabilitativo [6].

La tendenza ad utilizzare la riabilitazione computerizzata per il training cognitivo nella sclerosi multipla è in crescita [2] [7]. I programmi per la riabilitazione cognitiva forniscono esercizi focalizzati sui singoli domini cognitivi (attenzione, memoria, funzioni esecutive) e hanno evidenti vantaggi rispetto alle controparti cartacee, come ad esempio: la memorizzazione delle performance dei pazienti con grafici e tabelle di miglioramento; incremento della difficoltà automatico ed altre features decisamente utili nella riabilitazione. Oltre a software specifico per la riabilitazione sono stati provati alcuni giochi per il computer tagliati specificatamente per la riabilitazione cognitiva [11].

Esistono già alcune sperimentazioni circa la riabilitazione attraverso dispositivi mobili. Un gruppo di ricerca di Genova è riuscito a sviluppare una applicazione per la riabilitazione neuro cognitiva ottenendo una partecipazione dei pazienti molto alta (circa 84% dei pazienti coinvolti nello studio ha partecipato attivamente) e almeno l'81% tra loro era motivato a continuare ad utilizzare l'applicazione anche dopo la fine della sperimentazione [23]. Questi dati sono molto incoraggianti, nonostante il sistema utilizzato sia decisamente prototipale con schermate semplici e minimali, l'utilizzo di applicazioni mobili per la riabilitazione cognitiva apre la strada a un nuovo corso, un nuovo stile riabilitativo più vicino ai pazienti che possa inserirsi nella loro quotidianità avendo perciò un maggiore impatto nelle loro vite.

Alcuni tra i software per la riabilitazione cognitiva più popolari in Italia sono [15]:

Brainer: è un software *general purpose* basato su una piattaforma web per la ria-

bilitazione cognitiva sviluppato da Brainer srl (<https://www.brainer.it>). Brainer è stato usato da diverse cliniche e casi di studio, soprattutto su pazienti con schizofrenia, ma può essere sfruttato anche per il trattamento della sclerosi multipla.

Rehacom: è un software *general purpose* avanzato per la riabilitazione cognitiva distribuito da Hasomed (<https://www.rehacom.com>). Rehacom funziona su computer e richiede uno schermo di 19 pollici. Può essere integrato con una tastiera specifica. Rehacom propone molti esercizi sofisticati e basati su un contesto realistico tridimensionale supportando diverse lingue. Può essere sfruttato specificamente per la riabilitazione cognitiva della sclerosi multipla.

Erica: è un software *general purpose* per la riabilitazione cognitiva sviluppato da Giunti O.S. (<http://www.ericagiuntios.it>). Funziona su un normale computer. Anche Erica è stato usato per trattare la sclerosi multipla.

CogniPlus: è uno strumento sviluppato in ambito scientifico per il trattamento dei disturbi delle funzioni cognitive sviluppato da Schuhfried (<http://schuhfried.com>). CogniPlus è disponibile in 16 lingue. Ed è stato utilizzato in molti studi per trattare la sclerosi multipla.

1.2.4 Limiti dei sistemi esistenti

I sistemi presentati nel paragrafo precedente sebbene diffusi ed utilizzati per la riabilitazione nella sclerosi multipla non sono specifici per questa patologia. Sebbene Brainer offra più del doppio degli esercizi rispetto agli altri sistemi, questi non si basano su un contesto realistico, ancora più grave non vi è nessun tipo di adattamento della difficoltà degli esercizi durante la sessione di riabilitazione. La difficoltà può essere variata manualmente dall'operatore con soli tre gradi (facile, medio, difficile), caratteristica che rende il software poco fruibile da pazienti autonomi. Altra limitazione in Erica è data dal fatto che non è possibile fare esercizi a casa per i pazienti, tutte le sessioni riabilitative devono avvenire in ospedale. Altri software come Brainer sfruttano un approccio basato su architettura web mentre Rehacom e Cogniplus forniscono specifiche chiavette USB per la riabilitazione a domicilio.

Come mostrato dalla tabella 1.1 [15] i software descritti mostrano ancora molte mancanze per la riabilitazione cognitiva della sclerosi multipla, la mancanza di

	Brainer	Rehacom	Erica	CogniPlus
Target	General Purpose	General Purpose	General Purpose	General Purpose
Numero di esercizi	77	20+	35	15
Contesto realistico	no	si	no	si
Configurazione rapida	si	no	no	no
Esercitazioni a casa	si	si	no	si
Adattamento automatico	no	si	si	si
Supporto per i gruppi	no	no	no	no
Controllo remoto	no	no	no	no
Supporto ai tablet	si	no	no	no
Piattaforma web	si	no	no	no

Tabella 1.1: Comparazione tra i principali software per la riabilitazione cognitiva

esercizi specifici per la patologia, l'assenza di funzioni per la riabilitazione in gruppo e un difficile accesso alla riabilitazione domestica sono un freno alla diffusione di queste piattaforme. Si è perciò reso necessario studiare software riabilitativi specifici per il trattamento della sclerosi multipla che potessero coprire tutte le caratteristiche presentate e che offrissero un pacchetto riabilitativo completo multi ambito per i pazienti e per gli operatori.

1.3 Il sistema MS-Reahb

Tenendo conto di questa situazione è nata una collaborazione tra i neurologi e gli psicologi dell'ospedale Bellaria guidati dal dott. Sergio Stecchi e il gruppo di ricerca del prof. Mauro Gaspari del DISI. Nell'ambito di questa collaborazione sono stati individuati i punti deboli ed i limiti dei sistemi esistenti ed è stato progettato e sviluppato un nuovo sistema specifico per la riabilitazione cognitiva nella sclerosi multipla denominato MS-Rehab. MS-Reahb si propone come una nuova frontiera nella riabilitazione cognitiva domestica ed ospedaliera tramite un architettura basata sul web. Si configura come una applicazione web di facile utilizzo offre agli operatori uno strumento versatile e potente per la gestione del processo riabilitativo. MS-Reahb permette di conservare le cartelle cliniche dei pazienti gestire diverse valutazioni neuropsicologiche e permette di memorizzare risultati di batterie di test diversificati per ogni funzione cognitiva riabilitata. Il sistema vuole avvicinarsi anche ai pazienti con un interfaccia minimale e gradevole che possa stimolare in loro il desiderio di svolgere gli esercizi riabilitativi e che risulti intuitiva. Gli esercizi proposti sono composti da immagini semplici e chiare, brevi descrizioni del compito da svolgere e eventuali esempi per assicurarsi che il paziente abbia ben chiaro l'obiettivo dell'esercizio. Ogni schermata è progettata per

poter essere fruita dal paziente in piena autonomia nella comodità della sua abitazione, è il sistema stesso ad incrementare la difficoltà degli esercizi assegnati al paziente al variare della sua abilità nel risolverli, con un modello dinamico che si prefigge di far lavorare il paziente al suo livello soglia, ovvero assicurandosi sempre che il paziente trovi impegnativa la sfida riabilitativa e non esegua esercizi di routine troppo semplici che non stimolino il suo recupero psicologico.

1.3.1 Sezione operatori

Il sistema è diviso in due sezioni distinte, una sezione è specifica per gli operatori ovvero per tutto il personale sanitario (psicologici, neurologi, fisioterapisti e psicologi) che possono orchestrare la riabilitazione dei pazienti da un pratico pannello di controllo, l'altra è studiata per i pazienti. La parte più innovativa è sicuramente la presenza di un profilo che integra tutti i dati del paziente: anagrafico, clinico e neuro-psicologico, con dettagli circa i test psicologici a cui si è sottoposto e eventuali referti medici. La suddivisione dei vari profili per ciascun paziente permette agli operatori un rapido accesso al percorso riabilitativo del paziente contestuale al programma per la sua riabilitazione:

profilo anagrafico: Nel profilo anagrafico è possibile memorizzare alcuni dettagli circa la vita del paziente, l'età della scolarizzazione, l'impiego lavorativo, stato civili e numero di figli.

profilo clinico: Nel profilo clinico sono memorizzate tutte le caratteristiche correlate alla sclerosi multipla, anno di diagnosi, tipologia, terapie in atto e referti delle risonanze.

profilo neuropsicologico: Quest'ultima sezione permette di memorizzare nel dettaglio informazioni sullo stato neuropsicologico del paziente con dati inerenti ad alcuni test somministrati al paziente come i risultati della Rao e del White test. È possibile memorizzare diverse istanze di questi test prima e dopo la riabilitazione.

Sempre nella sezione degli operatori è possibile visualizzare e modificare i gruppi, ovvero gli insiemi di pazienti che svolgono un percorso riabilitativo comune, tale approccio si è rivelato molto proficuo presso il Bellaria dove sono soliti riabilitare dalle 4 alle 8 persone in sessioni settimanali.

Sempre in questo pannello di controllo è possibile configurare gli esercizi riabilitativi sia per singolo paziente sia per gruppo, l'approccio scelto è funzionale e

legato alla semplicità. Al personale è consentito scegliere quali esercizi assegnare e un livello di difficoltà ad essi associato, *facile, medio, difficile*, questa scelta è una vera rivoluzione per quanto riguarda la configurazione delle sessioni riabilitative. Nella maggior parte dei sistemi infatti la configurazione iniziale può richiedere la definizione di diversi parametri a volte con una grande perdita di tempo, in MS-rehab, si è riusciti ad estrapolare un modello che permette agli operatori di configurare un esercizio in pochi click settando in automatico la maggior parte dei parametri. Gli operatori potranno rapidamente configurare ed assegnare gli esercizi ai pazienti senza incorrere in difficoltose operazioni che nella maggior parte rappresentano più un ostacolo che una risorsa per la riabilitazione computerizzata. Nonostante questa semplificazione in fase di assegnamento di un nuovo esercizio il sistema durante le sessioni riabilitative offre una grana piuttosto fine all'avanzamento dei livelli di difficoltà, fornendo uno strumento versatile e potente in mano al paziente.

Altro aspetto importante del sistema è la possibilità da parte degli operatori sanitari controllare lo svolgimento degli esercizi di un paziente e di modificare il livello di difficoltà in tempo reale. Quando il paziente entra in una sessione riabilitativa di gruppo o privata, all'operatore è sempre data l'opportunità di vedere in tempo reale i suoi progressi, con un indice di miglioramento rispetto agli esercizi svolti in precedenza. L'operatore può quindi decidere di incrementare la difficoltà con una grana piuttosto fine: nella maggior parte dei casi in uno spettro da 0 (più facile) a 10 (più difficile). Il paziente appena completerà l'esercizio in corso si troverà con un nuovo esercizio della difficoltà decisa dall'operatore e non dal sistema in automatico.

Infine vi è la possibilità di monitorare i progressi di ciascun paziente nelle diverse aree riabilite, riguardo ai differenti esercizi proposti, ad esempio quanti esercizi effettivamente sono stati svolti dal paziente e le relative performance ottenute. Nel dettaglio l'operatore potrà visualizzare grafici aggregati secondo queste tipologie:

- *panoramica generale*: In questa sezione è possibile visionare in un unico grafico le curve di miglioramento delle performance di ciascuna funzione riabilitata, vi sarà perciò una curva per l'attenzione, una curva per la memoria e una per le funzioni esecutive.
- *aggregati per funzione riabilitata*: I grafici di questo tipo sono più specifici e mostrano l'andamento della performance relativo ad una funzione cognitiva specifica.

- *esercizio*: È possibile anche visualizzare l'andamento della riabilitazione relativo ad un singolo esercizio in modo da poter capire le specifiche difficoltà di un paziente e prendere provvedimenti adeguati, in questa sezione si scende ad un diverso tipo di dettaglio ed oltre ai grafici delle performance e dei successi/fallimenti è anche presente un grafico più dettagliato sulle risposte giuste, sbagliate, omesse e le tempistiche di ciascun esercizio.

1.3.2 Sezione per il paziente

La sezione relativa ai pazienti è più semplice rispetto a quella degli operatori ed offre una visuale chiara e immediata sugli esercizi disponibili in una data sessione riabilitativa che sia personale o di gruppo. Il paziente può selezionare un esercizio e cominciare la sessione, l'esercizio deve essere stato in precedenza assegnato dall'operatore e pertanto al paziente è data la possibilità di scegliere solo nel sottoinsieme per lui configurato. Quando un paziente si accinge a svolgere un esercizio può richiedere al sistema una sessione di *Training* nel caso non si ricordi come svolgerlo in modo corretto. La sessione di *Training* non valuta il paziente e offre istruzioni più dettagliate su come eseguire l'esercizio in modo corretto.

1.3.3 Esercizi disponibili

Gli esercizi ormai più di 20 offrono un panorama piuttosto ampio di alternative con esercizi specifici su differenti ambiti. Tra gli esercizi di attenzione figurano: *selettiva 1*, *selettiva 2*, *alternata*, *divisa*. Ciascuno di questi esercizi può essere svolto, con diversi tipi di stimoli, su *figure*, *volti* o *figure orientate (frecce o bussola)*. Tra gli esercizi di memoria vi sono: *riconoscimento*, *visuo-spaziale* e *di lavoro*. Di questi esercizi esiste una versione con *figure* una con *volti* e una con *figure orientate*. Infine vi sono gli esercizi legati alle funzioni esecutive, meglio descritte, in questo progetto, nella sezione progettazione. Tutti gli esercizi sono configurati per funzionare su tablet e computer con una particolare attenzione verso la componente tablet, il sistema vuole infatti offrire uno strumento utile agli ospedali e ai pazienti per la riabilitazione domestica, basata su hardware non dedicato allo scopo. Questo permette un abbattimento dei costi di sviluppo delle periferiche e favorisce una maggior diffusione del prodotto.

1.3.4 Realizzazione

MS-rehab è una applicazione web sostenuta da Apache Tomcat che implementa un Java Servlet e un Java Servlet Pages (JSP) da specifiche di Oracle Corporation

realizzata principalmente dal Dott. Floriano Zini del Disi con alcuni miei contributi. L'architettura di MS-rehab, e le tecnologie da esso sfruttate sono presentate in figura 1.3. Il sistema può essere utilizzato sia da tablet che da computer. Alcuni servizi tra i quali il monitoraggio individuale o per gruppi possono essere usufruiti anche via smartphone sebbene non sia questo il device di riferimento. L'interfaccia adattabile ai dispositivi è basata su Bootstrap, un famoso framework per sviluppare applicazioni *responsive* basato su tecnologie HTML, CSS e javascript. In questo modo l'applicazione può essere fruita su molti dispositivi eterogenei mantenendo compatibilità e coerenza adattandosi alle caratteristiche di dimensione e risoluzione dello schermo del dispositivo. Il server web si appoggia su una macchina virtuale con sistema operativo Ubuntu 16.04 che fornisce supporto a Java 8 e Apache Tomcat 8.0.39. Un database SQLite si preoccupa della persistenza dei dati circa gli utenti e le performance da essi ottenute. Al momento il server di MS-Rehab è ospitato al dipartimento di Scienze Informatiche dell'università di Bologna e può essere raggiunto all'indirizzo: <http://rehab.cs.unibo.it/MS-rehab-website/>. Sfruttando una nota e robusta architettura web, più centri di riabilitazione possono accedere al sistema e alle relative tecnologie disponibili. Inoltre sfruttando tecnologie ben note sarà possibile in caso di necessità migrare il sistema o replicarlo su differenti macchine per migliorarne prestazioni o affidabilità [15]. Il mio contributo durante il tirocinio presso il Disi si è articolato in due parti, memorizzazione delle performance dei pazienti negli esercizi svolti e successivamente sviluppare la componente che permettesse agli operatori di visualizzare grafici e istogrammi delle performance memorizzate. Infine ho sviluppato l'incremento manuale della difficoltà ad opera dei medici, grazie a questo sistema agli operatori sanitari è data la possibilità di alterare la difficoltà di un esercizio durante una sessione riabilitativa, sia ospedaliera che remota.

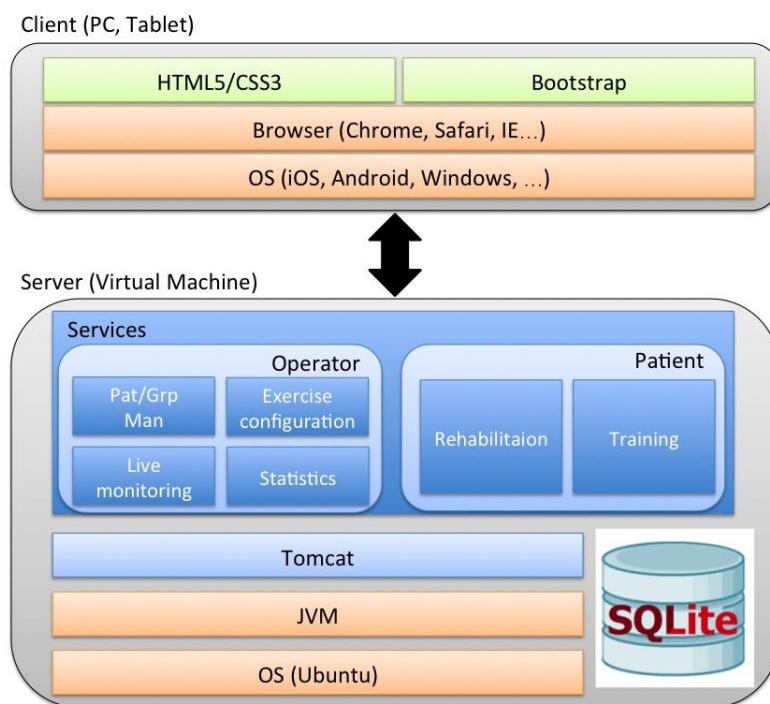


Figura 1.3: L'architettura web di *MS-Rehab*

Capitolo 2

Funzioni esecutive

Le funzioni esecutive si riferiscono a tutte quelle abilità coinvolte nella pianificazione, creare sequenze di azioni al fine di risolvere problemi o portare avanti attività; queste includono anche, organizzazione e regolazione del comportamento. Più in generale si può dire che le funzioni esecutive rappresentino quell'insieme di operazioni necessarie a portare a buon fine un compito. Senza una opportuna armonizzazione di tutte le funzioni cognitive non saremmo in grado di portare a termine nessun obiettivo della nostra vita, le funzioni esecutive sono lo strumento atto ad orchestrare le iterazioni necessarie al raggiungimento di un fine. Un deficit in queste funzioni cognitive ha un impatto immediato e visibile nella qualità della vita dei pazienti [16].

In questo capitolo viene descritto dettagliatamente il dominio cognitivo delle funzioni esecutive. Vengono inoltre presentate alcune tecniche generiche per la riabilitazione, descritti i dettagli della riabilitazione *ecologica* e infine, presentati alcuni esercizi specifici per questo dominio cognitivo sia con tecniche riabilitative tradizionali sia con esercizi digitali.

2.1 Esempi di funzioni esecutive

Le funzioni esecutive perciò riguardano tutti quegli aspetti che concorrono all'adattamento dell'uomo nell'ambiente circostante. Tra questi figurano [5]:

- capacità di pianificazione, ovvero saper formulare un piano generale e organizzare le azioni da intraprendere in una sequenza gerarchica;
- risoluzione dei problemi, ovvero come variare un piano formulato per far fronte ad una incombenza imprevista, o come formulare un piano logico

per risolvere un problema quotidiano, questa competenza riguarda anche i problemi matematici;

- capacità di astrarre, per mettere in risalto componenti essenziali di un problema da quelli trascurabili;
- la capacità di previsione sulle conseguenze delle azioni;
- l'immaginazione;
- la flessibilità nell'adattare l'uso delle capacità cognitive a seconda delle situazioni, ad esempio tra gli esercizi che riabilitano questa capacità vi è il compito di riconoscere i gatti tra le figure di animali e all'occorrenza di un suono il cominciare a riconoscere i cani.
- attivazione e riconoscimento di strategie appropriate alla risoluzione di un compito;
- capacità di esprimersi in maniera adeguata al linguaggio richiesto, utilizzando un linguaggio forbito in una lettera e colloquiale in una chat;
- acquisire consapevolezza del proprio stato cognitivo (metacognizione), e anche di quello degli altri individui, sia di riconoscerne il loro stato emotivo sia le loro aspirazioni desideri obiettivi;
- capacità di gestire gli aspetti emotivi della socializzazione, sia personali sia altrui.

Alcune di queste capacità richiedono ovviamente il coinvolgimento di altri domini cognitivi, l'attenzione e la memoria giocano un ruolo fondamentale nelle funzioni esecutive.

2.2 Riabilitazione delle funzioni esecutive

Sebbene i meccanismi che regolano lo svolgimento delle funzioni esecutive siano compresi ma non del tutto chiari, è stato evidenziato come queste risiedano in zone specifiche del cervello, ovvero nel lobo frontale della neocorteccia. Nella sclerosi multipla queste zone possono essere danneggiate in maniera selettiva, pertanto alcune funzioni esecutive possono risultare più compromesse di altre, e questo richiede l'attuazione di protocolli di riabilitazione altamente individualizzati. Questo complica le cose proprio perché non essendo ben chiari i meccanismi di

funzionamento, spesso le tecniche di riabilitazione utilizzate in letteratura hanno una scarsa generalizzazione in contesti di vita reale, in cui la risoluzione di compiti complessi può richiedere l'ausilio di strategie sempre diverse anche a fronte di minime variazioni. Alcune delle tecniche per la riabilitazione dei deficit nella risoluzione di problemi, tendono infatti ad educare il paziente circa le varie fasi che costituiscono questo procedimento complesso: l'identificazione del problema, la separazione dei dati rilevanti, il riconoscimento delle relazioni tra i diversi aspetti del problema, la produzione di strategie di risoluzione e il monitoraggio della bontà delle soluzioni generate [18]. Queste varie fasi vengono spesso riabilite separatamente mediante esercizi specifici, ad esempio per la fase di generazione delle soluzioni si può chiedere al paziente di completare delle storie che risolvono problemi di varia natura, e per la valutazione delle soluzioni si può coinvolgere il paziente nella attività riabilitativa di un altro paziente. Altri meccanismi che possono essere adottati per la risoluzione di problemi sono l'impiego di giochi di enigmistica, come ad esempio sudoku o i cruciverba. Un esercizio adottato per la riabilitazione delle funzionalità di pianificazione è quello di descrivere i passi necessari a svolgere un dato compito, ad esempio la preparazione di una ricetta. Altro approccio per riabilitare il paziente nelle funzioni esecutive può essere quello di sviluppare problemi legati al problem-solving in cui vi sono tappe intermedie che apparentemente allontanano dalla meta finale, come ad esempio la risoluzione di labirinti o esercizi basati su schemi analoghi. In quest'ottica risulta molto utile anche l'impiego di giochi di strategia guidati da regole in cui il paziente deve intraprendere delle strategie che devono rispettare vincoli, come ad esempio pianificare le mosse degli scacchi dove ogni pezzo ha alcuni vincoli di movimento sulla scacchiera. Spesso la difficoltà di stabilire l'efficacia di un trattamento riabilitativo delle funzioni esecutive consiste nel fatto che è molto difficile valutare in maniera oggettiva le prestazioni del paziente, vista l'assenza di protocolli di valutazione preliminari per molte delle funzioni esecutive elencate.

2.3 Riabilitazione ecologica

Gli esercizi riabilitativi, siano essi specifici per le funzioni esecutive o per altri domini, si dividono in due categorie esercizi astratti e esercizi realistici. Gli esercizi realistici nella maggior parte dei casi permettono una riabilitazione *ecologica* del paziente. Per riabilitazione ecologica si intende un insieme di attività che allenino il paziente con disabilità intellettiva su compiti tipici richiesti nella vita quotidiana.

Essendo le funzioni esecutive di prioritaria importanza nella vita quotidiana dei pazienti è fondamentale che gli esercizi atti alla loro riabilitazione siano quanto

più possibili realistici e inseriti in un contesto credibile. Soltanto in questo modo tali esercizi riusciranno ad impattare sulla qualità della vita dei pazienti in modo positivo.

2.4 Esempi di esercizi riabilitativi

Uno degli esercizi più noti appartenente alla categoria (BADS-C) è sicuramente la *Mappa dello Zoo*, un esercizio specifico per la riabilitazione delle funzioni esecutive, proposto da Alderman et al., nel 1996. Al paziente viene richiesto, data la mappa di uno Zoo con alcune possibili punti di interesse, solitamente 12, di visitarne un sottoinsieme, normalmente 6. Questo esercizio è atto a misurare le capacità di *problem solving* e pianificazione attraverso l'abilità nella messa in atto di un comportamento sequenziale. All'esercizio sono presenti alcune difficoltà successive primo tra tutti il vincolo dell'ordine con il quale visitare i luoghi e in secondo luogo vincoli sulle strade che possono essere transitabili una volta sola o in una sola direzione. Questi vincoli rendono più complesse le soluzioni che il paziente deve produrre per l'obiettivo.

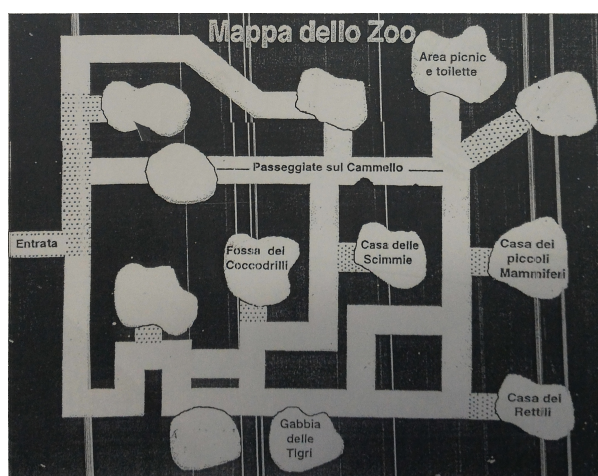


Figura 2.1: Esempio di un esercizio della *Mappa dello zoo*

L'esercizio che è stato automatizzato in questa tesi è chiamato *Mattina di impegni (Plan a day)*, proposto da Funke, J. & Krüger, T. (1993) [13], una versione più complessa rispetto allo Zoo, al paziente viene fornita una serie di compiti da svolgere in una giornata non ordinati, come ad esempio andare in banca a prelevare il denaro, portare la figlia in piscina e degli orari ai quali svolgere suddetti impegni. Il compito del paziente è pianificare una sequenza di azioni che gli consenta di

fare tutto quanto nei tempi stabiliti. L'esercizio richiede anche di effettuare delle inferenze logiche piuttosto interessanti, il baule dell'automobile pieno deve essere svuotato per far fronte al volume della spesa per 3 persone richieste dall'esercizio. Si tratta di passaggi da effettuare che denotano un livello piuttosto elevato di comprensione della realtà e richiedono alcune strategie di azione molto stimolanti per pazienti affetti da sclerosi multipla.

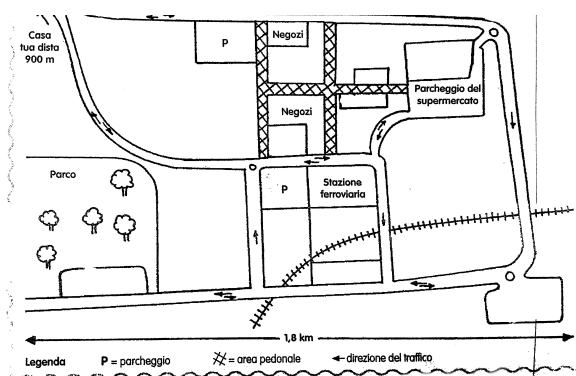


Figura 2.2: Esempio di un esercizio della *Mattina di impegni*

Un altro esempio utilizzato presso il Bellaria è chiamato *Mappa di Firenze* molto simile alla *Mattina di impegni* e alla *Mappa dello Zoo*: al paziente è richiesto di visitare una serie di monumenti appartenenti alla città di Firenze quali la Galleria degli Uffizi il Duomo, il Museo di S. Marco ed altri ancora cercando di fare tutto in un tempo fissato prestando attenzione che in alcuni musei è richiesta una visita di almeno 2 ore.

In tutti gli esercizi presentati le mappe sono stereotipate, rappresentate secondo una versione semplificata, con poche strade dove i collegamenti sono funzionali a collegare gli edifici principali. Le mappe mettono in luce solo quegli edifici necessari a rappresentare posizioni utili per la navigazione dell'utente. Gli esercizi sono inoltre svolti tramite carta e penna dove il paziente è chiamato a raccontare le sue intenzioni circa gli spostamenti che intende compiere e l'operatore può seguire passo passo lo svolgimento dell'esercitazione.

2.5 Realizzazioni attuali

Alcuni software, tra quelli che abbiamo illustrato nella sezione 1.2.3, includono realizzazioni di questi esercizi. In *CogniPlus* è presente un esercizio per la riabilitazione delle funzioni esecutive, questo esercizio è una ulteriore evoluzione della

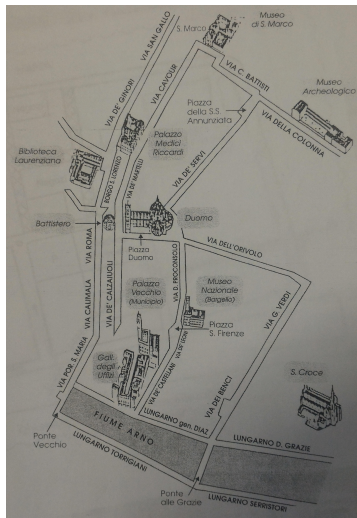


Figura 2.3: Esempio di un esercizio della *Mappa di Firenze*

versione di *Plan-a-Day* proposta nel 1993 (Figura 2.4 sinistra), la sua interfaccia è piuttosto minimale con i *task* presentati sulla destra e una mappa stilizzata sulla sinistra (Figura 2.4 destra). I *task* sono semplici e si ripetono nel tempo, la mappa rimane invariata. Non esiste l'incremento automatico della difficoltà durante lo svolgimento dell'esercizio e il paziente deve comprendere l'ordine di risoluzione dei singoli *task* per concludere l'esercizio correttamente. Il programma non offre supporto alla lingua Italiana.

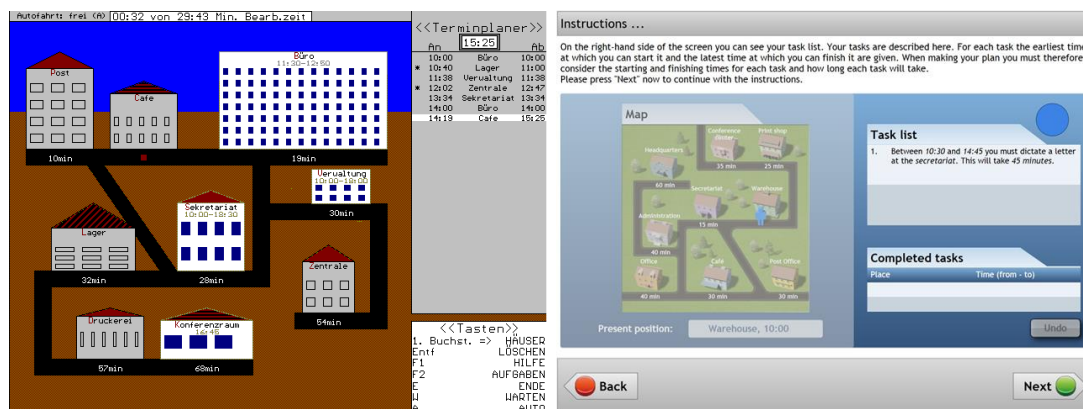


Figura 2.4: Esempio di un esercizio di *Plan-a-Day* di *CogniPlus*

Nonostante i limiti, questo software è stato utilizzato in molte sperimentazioni di riabilitazione cognitiva in pazienti affetti da schizofrenia [21], con miglioramenti nelle funzioni cognitive e nelle capacità funzionali.

Capitolo 3

Planning e PDDL

La pianificazione, ovvero il processo di pensare e organizzare le attività necessari al raggiungimento di un obiettivo, è un compito complesso soprattutto in ambito informatico. La pianificazione riguarda un numero estremamente ampio di compiti e richiede strategie diversificate per risolvere ciascuno di essi in modo adeguato, è possibile parlare di pianificazione di spostamento, e pianificazione delle traiettorie, nel campo delle intelligenze artificiali i compiti di pianificazione possono variare dal risolvere un puzzle come il cubo di Rubik, risolvere il gioco del 15 o costruire una pila di blocchi. Storicamente gli algoritmi di *planning* vengono considerati differenti da quelli di *problem solving*, ciononostante le distinzioni sono via via andate a scemare negli anni.

In questo capitolo si è cercato di descrivere in generale i problemi di pianificazione con maggiore dettaglio al linguaggio chiamato PDDL, infine si è approfondito il suddetto linguaggio con esempi del suo funzionamento.

3.1 Pianificazione

Per definire correttamente cosa sia un problema di pianificazione occorre scomporlo nei suoi elementi basilari [17]:

- **Stato** I problemi di pianificazione riguardano uno spazio degli stati che cattura tutte le possibili situazioni che si possono verificare. Lo stato potrebbe, ad esempio, rappresentare la posizione e l'orientamento di un robot, la posizione delle tessere in un puzzle. È possibile avere uno spazio degli stati discreto o continuo. Nella maggior parte dei casi lo spazio degli stati è troppo grande per poter essere computato interamente o rappresentato in modo esplicito. Nonostante questo lo spazio degli stati è un astrazione fon-

damentale per poter trattare la complessità di un algoritmo di pianificazione e costruire strategie risolutive adeguate.

- **Tempo** Tutti i problemi di pianificazione riguardano una sequenza di decisioni che devono essere effettuate nel tempo. Il tempo può essere esplicito nel modello, come guidare una automobile rapidamente nel traffico. Il tempo può anche essere implicito come decidere la più breve successione di mosse per risolvere il cubo di Rubik.
- **Azioni** Un piano è una sequenza di azioni che manipolano gli stati. Le azioni rappresentano una funzione di transizione tra stati del problema.
- **Stato iniziale e stati obiettivo** Un algoritmo di pianificazione solitamente riguarda uno stato iniziale e cerca un piano per raggiungere lo stato finale (o un insieme di stati finali).
- **Criterio** Raggiungere lo stato finale può avvenire secondo differenti strategie definire un criterio con il quale raggiungerle è spesso parte essenziale dei problemi di pianificazione. Possibili criteri sono:
 1. **Fattibilità** trovare un piano qualsiasi che raggiunga lo stato obiettivo a prescindere dal costo.
 2. **Ottimalità** trovare un piano ottimale secondo alcuni parametri che raggiunga lo stato obiettivo. Nella maggior parte dei casi questo si traduce in un problema di ottimizzazione.

3.2 PDDL

Vista la complessità degli algoritmi di pianificazione Drew McDermott e i suoi colleghi svilupparono un nuovo standard per la pianificazione nell'ambito delle Intelligenze Artificiali chiamato *Planning Domain Definition Language (PDDL)* basato sugli algoritmi *STRIPS* e *ADL*. Il linguaggio si basa su un modello della pianificazione costituito da due parti: la descrizione del dominio e la relativa descrizione del problema. Questa divisione semplifica la formalizzazione di nuovi problemi per un dominio già noto. Infatti tutti quegli aspetti comuni ad una classe di problemi devono essere formalizzati nella descrizione del dominio quali ad esempio le *azioni* e gli *stati*. Nella descrizione del problema vengono invece specificati le *condizioni iniziali e finali*, gli *stati permessi* e eventuali *criteri* per individuare una soluzione [1].

3.2.1 Modellare con PDDL

Per modellare il dominio occorre definire alcuni aspetti fondamentali, si è utilizzata la terminologia inglese perché corrisponde a specifiche parole chiave:

domain: Il nome del dominio;

requirement: In PDDL è necessario dichiarare quale aspetto del modello che il sistema deve sfruttare come ad esempio termini atomici, uguaglianze, tipizzazione etc..;

object-type hierarchy: Definire una gerarchia di entità, ovvero come gli elementi sono posti in relazione, ad esempio *macchina* appartiene alla classe dei *veicoli*;

constants: Definizione di quegli elementi costanti a tutta la classe di problemi associata al dominio;

predicates: Atto a definire un insieme di fatti logici deducibili;

actions: Un insieme di operatori che agiscono sugli stati, ciascuna azione è composta da alcune parti fondamentali:

parameters: Parametri necessari affinché l'azione possa essere compiuta, rappresentati dalle entità del modello;

preconditions: Rappresentano in quali condizioni devono trovarsi le entità del modello affinché possa essere intrapresa una certa azione;

effects: Ovvero come variano le entità del modello al termine dell'azione svolta.

Le azioni possono presentare anche operazioni condizionali **when-effects**.

Quanto descritto viene poi implementato secondo la sintassi di PDDL di cui si presenta un esempio in Listing 3.1. La figura mostra due predicati linea 8 e 9 il primo predicato *at* accetta un oggetto *?x* e una posizione *?l* il punto di domanda rappresenta le variabili in questo formalismo. Vi è poi a linea 11 un azione che specifica che qualora l'oggetto *B* si trovi in *?m* e *?m* diverso da *?l* allora verrà applicato l'effetto a riga 14, ovvero sposta l'oggetto *B* in *?l*.

```

1 (define (domain briefcase-world)
2   (:requirements :strips
3     :equality
4     :typing
5     :conditional-effects)
6   (:types location physob)
7   (:constants (B - physob))
8   (:predicates (at ?x - physob ?l - location)
9     (in ?x ?y - physob)
10  )
11  (:action move-b
12    :parameters (?m ?l - location)
13    :precondition (and (at B ?m) (not (= ?m ?l)))
14    :effect (and (at B ?l) (not (at B ?m)))
15  )
16 )
17 ...
18 )

```

Listing 3.1: "Un esempio di dominio"

Per modellare un problema è necessario invece definire i parametri fondamentali necessari al raggiungimento della soluzione:

problem: Il nome del problema;

domain: Il dominio relativo al problema;

objects: Ovvero gli elementi presenti nell'universo, questi sono *atomici* e locali all'universo definito, possono essere inseriti in una gerarchia;

init: Un insieme di stati che rappresentano lo stato di partenza del sistema, queste condizioni possono essere congiunzioni di valori booleani vero o falso;

goal: Un insieme di stati finali, un insieme di elementi che devono essere veri o falsi al termine della computazione.

```

1 (define (problem briefcase-p)
2   (:domain briefcase-world)
3   (:objects P D)
4   (:init
5     (place home)
6     (place office)
7     (at B home)
8     (at P home)
9     (at D home)
10    (in P)
11  )
12  (:goal (and
13    (at B office)
14    (at D office)
15    (at P home)
16  )
17 )
18 )

```

Listing 3.2: "Un esempio di problema"

L'esempio presentato in Listing 3.2 mette in luce tutti gli aspetti tipici di un problema come i goal uniti da una congiunzione in `and` e gli stati iniziali ovvero i predicati noti al problema.

Dal 1998 a oggi sono state sviluppate molte migliorie a questo modello iniziale con PDDL2.1 sono stati inseriti i *numeric fluents* ovvero operatori non binari che rappresentano elementi del modello. Il sistema è poi cresciuto negli anni per contributi di altri autori riuscendo a risolvere problemi multi agente, in intervalli temporali discreti e continui.

Capitolo 4

Il pianificatore

4.1 PDDL applicato al processo risolutivo

Al fine di sviluppare un pianificatore in grado di risolvere un esercizio del tipo *Mattina di impegni* sono state estrapolate le azioni atomiche necessarie a risolvere il problema:

- Muoversi al fine di raggiungere un luogo
- Rimanere per un certo numero di ore in un luogo
- Acquistare (prendere) un oggetto
- Rilasciare (lasciare) un oggetto
- Scambiare un oggetto (scambiare denaro con caffè)
- Finire in un luogo specifico

ciascuna di queste azioni è poi stata formalizzata in una azione del pianificatore come descritto in questo capitolo.

Ed i seguenti vincoli:

- Rispettare gli orari di apertura di un luogo
- Rispettare gli orari di chiusura di un luogo
- Attraversare un percorso pedonale o riservato alle automobili
- Iniziare da un luogo specifico

In fase di progettazione, sono stati individuate in maniera informale le entità coinvolte nel sistema PDDL, a tal fine possiamo classificarle come segue:

tempo: vincoli temporali espressi mediante la sequenza $p_1 .. p_{48}$, ciascun p_i rappresenta 30 minuti;

place: rappresentano i luoghi raggiungibili durante lo svolgimento dell'esercizio;

waitplace: rappresentano una sottocategoria di luoghi nei quali è possibile aspettare;

item: rappresentano gli oggetti che possono essere raccolti negli edifici;

vehicle: rappresentano uno specifico item che può essere sfruttato come veicolo.

Formalizzare le entità coinvolte è stato necessario definire i predicati che creano relazioni tra tali entità (Listing 4.1):

```
1 (: predicates
2     (now ?t – tempo)
3     (next ?t1 – tempo ?t2 – tempo)
4     (past ?t – tempo)
5     (future ?t – tempo)
6     (be ?src – location)
7     (path ?src – location ?dst – location)
8     (road ?src – location ?dst – location)
9     (have ?pos – location ?item – item)
10    (have-in-change ?pos – location ?item – item ?change – item)
11    (got ?item – item)
12    (got-in-change ?change – item ?item – item)
13    (open-hour ?pos – location ?t – tempo ?tc – tempo)
14    (first-number ?t – tempo)
15    (stand-1h ?pos – waitplace)
16    (stand-2h ?pos – waitplace)
17    (stand-3h ?pos – waitplace)
18    (stand-4h ?pos – waitplace)
19    (be-at ?src – location ?t – tempo)
20    (drop-at ?item – item ?pos – location ?t – tempo)
21    (got-at ?item – item ?pos – location ?t – tempo)
22    (can-drop ?item – item ?pos – location)
23 )
```

Listing 4.1: "L'insieme dei predicati necessari a definire un problema"

Infine ci si è concentrati sul risolutore di problemi, ovvero sulla sequenza di azioni abilitanti a risolvere un problema dato e descritto tramite PDDL. A tal fine si è cercato di formalizzare le azioni individuate in precedenza secondo la sintassi del *solver*.

```

1 (:action move-to
2   :parameters (
3     ?src – place
4     ?dst – place
5     ?t0 – tempo
6     ?t1 – tempo)
7   :precondition (and
8     (be ?src)
9     (path ?src ?dst)
10    (now ?t0)
11    (next ?t0 ?t1)
12    (future ?t1)
13  )
14  :effect (and
15    (not (be ?src))
16    (be ?dst)
17    (not (now ?t0))
18    (not (future ?t1))
19    (now ?t1)
20    (past ?t0)
21    (be-at ?dst ?t1)
22  )
23 )

```

Listing 4.2: "L'azione base di spostamento tra un luogo e l'altro"

Ad esempio l'azione espressa in Listing 4.2 è piuttosto semplice da descrivere per poter essere attivata è necessario che il paziente si trovi in *?src* ed esista una strada tra *?src* e *?dst* se tale è vero e l'orario è *?t₀*, e la mezz'ora successiva è *?t₁* allora le precondizioni sono accettate. Perciò gli effetti possono essere attivati, in questo caso il paziente non si troverà più in *?src* ma bensì sarà in *?dst* e il tempo sarà passato da *?t₀* a *?t₁*, qui si aggiornano anche predicati di ambiente quali (*past ?t0*) ad indicare che ora *?t₀* appartiene al passato. Altro predicato importante è l'informazione che il paziente sia stato in *?dst*.

Come è evidente è stato necessario modellare il tempo in modo esplicito poiché gestire un tempo continuo complica ulteriormente i vincoli del sistema rendendo

molto difficile trovare una soluzione in tempi macchina contenuti. La scelta è stata imposta anche dal *solver* utilizzato una versione di PDDL ibrida che non sfruttava completamente i predicati numerici.

```

1 (:action take-item-at
2   :parameters (
3     ?pos – place
4     ?item – item
5     ?tp – tempo
6     ?t – tempo
7     ?tc – tempo
8     ?tnow – tempo)
9   :precondition (and
10     (be ?pos)
11     (have ?pos ?item)
12     (open-hour ?pos ?t ?tc)
13     (past ?tp)
14     (next ?tp ?t)
15     (now ?tnow)
16     (future ?tc)
17   )
18   :effect (and
19     (not (have ?pos ?item))
20     (got ?item)
21     (got-at ?item ?pos ?tnow)
22   )
23 )

```

Listing 4.3: "L'azione base di prendere un oggetto in un luogo"

Per formalizzare il vincolo "*Acquistare (prendere) un oggetto*" in un certo luogo (Listing 4.3) sono stati necessari alcuni accorgimenti primo tra tutti memorizzare in un predicato l'orario dell'evento avvenuto. In questo senso possiamo notare che tra le condizioni oltre ad essere nello stesso luogo dell'oggetto ovvero trovarsi in *pos* e che l'oggetto sia in quella posizione (*have ?pos ?item*) bisogna assicurarsi che il negozio o l'ambiente sia accessibile a quell'orario ciò avviene in un modo piuttosto ingegnoso. Si è implementato tra i predicati un sistema ordinato di successori tramite il predicato (*next ?t_p ?t*) grazie a ciò è possibile vincolare gli orari di apertura del negozio. Infatti se il negozio apre in *?t* e chiude in *?t_c* possiamo essere certi di trovarci in un orario di apertura se:

$$\exists ?tp : ?tp \in \text{passato} \wedge ?tc \in \text{futuro} \wedge ?tp + 1 = ?t \Leftrightarrow \text{aperto} \quad (4.1)$$

Questo vincolo viene espresso bloccando a catena i predicati, purtroppo il solver non gestisce predicati con uguale valore ma nome differente, infatti $?t_{now}$ che rappresenta il tempo corrente non può assumere per come funziona PDDL lo stesso valore di t perciò risulta evidente che non è possibile accedere con questo vincolo ad un oggetto del negozio al suo orario di apertura per far fronte a questo problema si è aggiunta una seconda azione chiamata *take-item-at-now* che accorpa $?t$ con $?t_{now}$ e che funziona qualora le due variabili abbiano lo stesso valore.

Entrambe le azioni hanno come risultato di togliere l'oggetto dal luogo, ottenere l'oggetto designato e un predicato aggiuntivo che sancisce il momento nel quale si ha avuto suddetto oggetto. Tale predicato non viene mai rimosso dal sistema e serve a memoria per eventuali obiettivi correlati all'aver avuto un dato oggetto o fatto una data azione.

Con il medesimo principio di concatenamento dei vincoli si è riusciti anche ad esprimere il vincolo di attesa. Come mostrato in Listing 4.4 è evidente che il sopracitato meccanismo concatena il tempo presente con il tempo successivo. Si può notare anche che tra i parametri figura la presenza di *pos* qualificato come *waitplace* ciò forza il fatto che sia possibile attendere solo nei luoghi designati dal sistema. Grazie alla azione *still-1h* il paziente ottiene il predicato (*stand-1h ?pos*) ovvero l'aver atteso un dato ammontare di unità temporali nel luogo designato. Non potendo utilizzare i numeri nel nostro modello si è deciso di definire quattro attese possibili *stand-1h*, *stand-2h*, *stand-3h*, *stand-4h* che vanno rispettivamente da una attesa di 30 minuti fino ad una attesa di 2 ore. Per attese più lunghe è possibile concatenare più cicli di attesa.

```

1 (:action still-1h
2   :parameters (
3     ?pos – waitplace
4     ?t0 – tempo
5     ?t1 – tempo)
6   :precondition (and
7     (now ?t0)
8     (next ?t0 ?t1)
9     (future ?t1)
10    (be ?pos)
11
12    )
13   :effect (and
14     (not (now ?t0))
15     (not (future ?t1))
16     (now ?t1)
17     (past ?t0)
18     (stand-1h ?pos)
19     (be-at ?pos ?t1)
20   )
21 )

```

Listing 4.4: "L'azione base di attendere un unità temporale"

4.2 Mantenere generale il problema

Progettato il risolutore di problemi è stato necessario sviluppare una componente essenziale per il sistema, ovvero, un generatore di problemi. Va infatti considerato che uno dei principali requisiti di questo sistema è il costante apporto di nuovi problemi da presentare al paziente in modo che sia virtualmente impossibile che un paziente risolva due volte lo stesso. Ciò permette al paziente di allenarsi sulla pianificazione in generale e non di apprendere meccanismi specifici per risolvere una classe di problemi.

Per rendere il più generale possibile il generatore di problemi si è scelto di costruire un file *json* con una sintassi semplice che potesse rappresentare gli scenari in modo generico.

```

1 {
2   "version":1.8,

```

```

3   "places-list":["casa","banca","piscina","discarica","supermercato
    ","farmacia","gelateria","stazione"],
4   "item-list":["figlia","macchina","caffè","ombrello","
    "cellulare","orologio","denaro","bancomat","maschera","
    "tessera","rifiuti","mobili","spesa","frutta","verdura","
    "carne","bibite","pillole","sciroppo","collirio","gelato",
    "crepes","marco","biglietto","abbonamento"]
5 }

```

Listing 4.5: "Estratto di un esempio di json header"

Nell'header viene specificata la lista di luoghi presenti nello scenario e la lista di oggetti disponibili nello scenario.

```

1 {
2   "name":"piscina",
3   "img":"piscina",
4   "preposition":"in",
5   "article":"la",
6   "min-hour":16,
7   "max-hour":40,
8   "min-open-hour":12,
9   "wait":"true",
10  "items-take":[
11    "figlia",
12    "maschera",
13    "tessera"
14  ],
15  "items-leave":[
16    "denaro"
17  ],
18  "actions":[
19    "move-to",
20    "drive-to",
21    "take-item-at",
22    "take-item-at-now",
23    "change-item",
24    "drop-item-at",
25    "drop-item-at-now",
26    "still-1h",
27    "still-2h",

```

```

28     "be-at"
29   ]
30 }

```

Listing 4.6: "Estratto di un esempio di json lista luoghi"

Ciascun luogo viene poi descritto nel dettaglio, specificando nome e immagine line 2, 3 o il minimo numero di ore per cui deve rimanere aperto l'esercizio riga 6. In ogni stabile viene definito anche gli oggetti che è possibile trovare in quel luogo, linea 10, e gli oggetti che è possibile depositare in tale luogo (linea 15). Infine viene specificato quali azioni sono ammissibili in quel dato luogo (linea 18).

```

1 {
2   "name":"mobili",
3   "img":"mobili",
4   "article":"i",
5   "actions":[
6     "take-item-at",
7     "take-item-at-now",
8     "change-item",
9     "drop-item-at",
10    "drop-item-at-now"
11  ]
12 }

```

Listing 4.7: "Estratto di un esempio di json lista oggetti"

Come si evince chiaramente la struttura del file il *json* permette agilmente di scrivere nuovi ambienti specificando quali sono i luoghi *places-list* quali gli oggetti *item-list*. Inoltre per ciascuno di essi specifica quali azioni sono legittime e quali oggetti è legittimo trovare in quel luogo. Ad esempio si è detto che in piscina che è aperta al minimo dalle 8.00 fino alle 20.00, per un minimo di 6 ore (Il tutto espresso in mezz'ore) sia possibile attendere in piscina e ci sono alcuni oggetti che possono essere rinvenuti quali *figlia*, *maschera*, *tessera*: la figlia appare essere un oggetto per come è strutturato il sistema, questa rappresenta una semplificazione del mondo reale ma è quanto necessario per rendere veloce il sistema. Vi sono poi specificati gli elementi che possono essere lasciati in piscina in questo caso il *denaro* e le azioni che è possibile svolgere in piscina, scritte nel campo *actions*.

Ciò è stato fatto per creare in modo automatico mappe credibili e realistiche che non mettessero oggetti senza senso in un luogo come i *mobili* in piscina o il *gelato* in banca, ma allo stesso tempo lasciasse al sistema ampia autonomia su come costruire i problemi. Se infatti si volesse definire il problema dello zoo o

del viaggio a Firenze sarebbe sufficiente descrivere una nuova lista di posti e di oggetti e specificare in quali zone possono essere riposti, oggetti come la *cartolina* in una eventuale *edicola* oppure la *fotografia dei leoni* presso la *gabbia dei leoni*.

4.3 Il generatore

Il generatore di problemi è composto da due parti distinte:

- *generatore di mappe* si preoccupa di comporre un ambiente credibile ma casuale con edifici, rotonde e oggetti.
- *generatore di problemi* questo secondo generatore è strettamente collegato al primo e data una mappa genera un problema risolvibile in quella mappa.

Per quanto riguarda la generazione delle mappe si è scelto un approccio semi casuale, con alcune linee guida dovute al file json precedentemente descritto. Vengono estratte un numero casuale di edifici ed aggiunte le rotonde e vengono disposti su una griglia 3*2 o 4*2 o 4*3 nel caso massimo. Gli edifici vengono poi collegati con strade principali e successivamente vengono aggiunti dei collegamenti random tra edifici vicini. Per ciascun edificio vengono aggiunti alcuni oggetti presenti tra i suoi disponibili fino ad un massimale. Questo approccio assicura la massima casualità nella struttura delle città senza però rinunciare alla coerenza. Successivamente in base alla mappa vengono generati degli obiettivi suddivisi in due macro aree.

- *obiettivi legati a posti*: Tutti quegli obiettivi correlati allo spostamento o al sostare in un dato ambiente, figurano tra questi *be*, *be-at*, *stand*;
- *obiettivi legati agli oggetti*: Tutti quegli obiettivi correlati al ottenere, rilasciare, scambiare oggetti di varia natura, figurano tra questi *got*, *got-at*, *drop-at*.

Viene deciso a priori da quanti sotto obiettivi l'esercizio è composto. Viene pescato a caso un predicato ovvero un obiettivo di PDDL e vengono sulla base della mappa popolati i suoi campi. Se ad esempio l'obiettivo è legato ad un oggetto come *got-at* il generatore filtra tra tutti gli oggetti aggiunti alla mappa quelli che possono venire raccolti, e successivamente sulla base di dove si trova l'oggetto sceglie un orario consono per raccogliarlo ovvero entro gli orari di apertura e chiusura dell'edificio. Il problema così generato viene analizzato dal *solver* che determina se sulla base del vincolo aggiunto l'esercizio è ancora soddisfacibile se

lo è viene scelto a caso un nuovo predicato e si continua fino a quando il problema non ha raggiunto la complessità richiesta. Questo processo ovviamente può generare problemi molto lunghi e complessi anche con pochi obiettivi, immaginiamo che per ritirare il latte sia necessario attraversare tutta la città aspettare molte ore e magari scambiare il latte con un oggetto che si trova da tutta altra parte. Perciò i problemi così ottenuti vengono classificati in 3 categorie (facile, medio, difficile) per complessità sulla base del numero di step che il *solver* trova nella soluzione migliore, con un piccolo incremento correlato con la dimensione della mappa, problemi semplici possono sembrare più complicati su mappe più grandi.

Capitolo 5

La progettazione dell'applicazione

Una volta realizzato il pianificatore, si è proceduto con l'integrazione dell'esercizio in MS-Rehab. Questo ha comportato una serie di problematiche affrontate in questo capitolo che tratta l'analisi dei requisiti messi in luce dai casi d'uso, i diagrammi di sequenza per descrivere lo svolgimento dei task presentati nei casi d'uso, i diagrammi delle classi per spiegare lo stato del sistema prima e dopo le modifiche, le scelte grafiche ed estetiche compiute nel design delle interfacce e infine la valutazione delle performance dei pazienti.

5.1 Casi d'uso

Durante la fase dell'analisi dei requisiti si è vista necessaria la stesura di un diagramma dei casi d'uso per comprendere meglio le funzionalità del prodotto software e chiarificare al meglio quali componenti grafiche sarebbero state necessarie successivamente. I casi d'uso individuati sono descritti nella figura in seguito e formalizzati nel paragrafo sottostante:

CREA NUOVO SCENARIO

Descrizione: Consente al personale sanitario autenticato di generare un numero arbitrario di nuovi scenari.

Attori: Il personale sanitario, medici, psicologi, logopedisti.

Flusso principale: Il caso d'uso si ha quando l'utente visita la pagina atta a generare nuovi esercizi e preme sull'apposito controllo.

Post-condizioni Applicazione permanente delle operazioni sul database

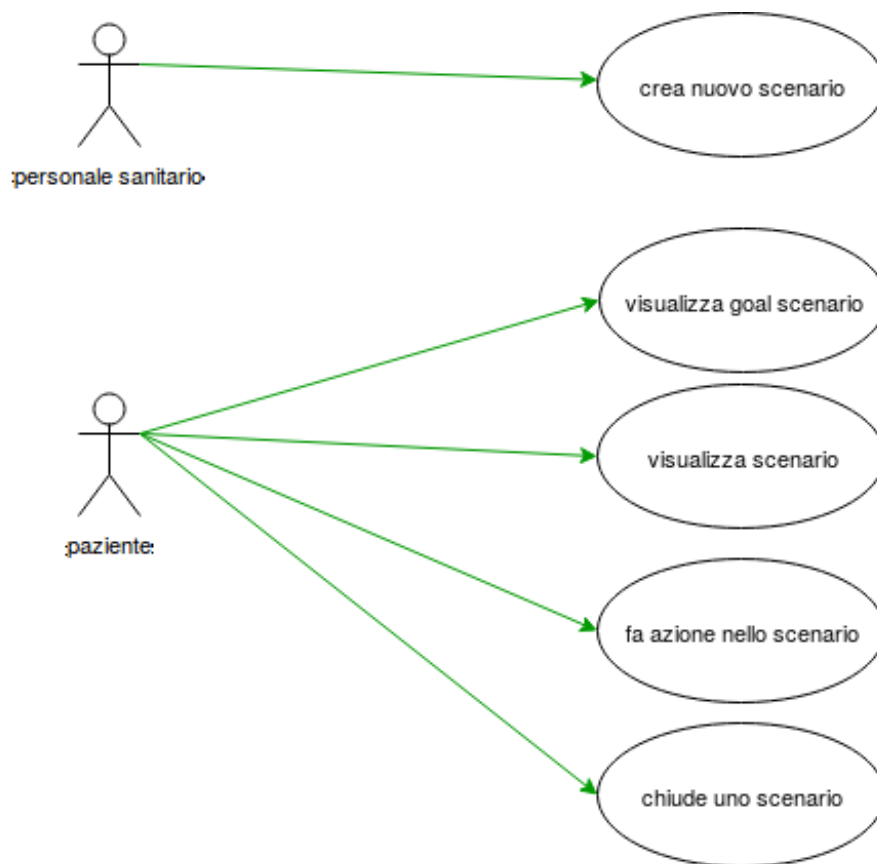


Figura 5.1: Diagramma dei casi d'uso

Questo caso d'uso rappresenta la più basilare interazione degli operatori con il sistema per renderli autonomi nella generazione di un numero sempre nuovo di esercizi.

VISUALIZZA GOAL SCENARIO

Descrizione: Consente al paziente autenticato di visualizzare gli obiettivi di un esercizio.

Attori: Il paziente autenticato.

Flusso principale: Il paziente visualizza un nuovo esercizio con gli obiettivi assegnati e può procedere verso lo svolgimento dell'esercizio.

Post-condizioni Nessuna

Questo caso d'uso rappresenta per il paziente la possibilità di visualizzare gli obiettivi dell'esercizio in qualsiasi momento durante il suo svolgimento.

VISUALIZZA SCENARIO

Descrizione: Il paziente autenticato visualizza l'ambiente dell'esercizio con il quale può interagire

Attori: Il paziente autenticato.

Flusso principale: Il paziente visualizza il nuovo scenario e può interagirvi.

Postcondizioni Nessuna

Questo caso d'uso rappresenta la possibilità per il paziente di avviare un esercizio.

FA AZIONE NELLO SCENARIO

Descrizione: Il paziente autenticato esegue un azione nello scenario proposto.

Attori: Il paziente autenticato

Flusso principale: Il paziente muove con mouse o interfaccia touch la pedina sullo schermo per compiere una azione di movimento o preme sui bottoni atti a raccogliere un oggetto o rilasciare o scambiare un oggetto.

Postcondizioni Il sistema calcola se l'azione è ammissibile e restituisce il controllo al paziente.

Questo caso d'uso rappresenta una classe estesa di azioni tra le quali spostarsi, raccogliere oggetti e in generale interagire con l'esercizio.

CHIUDE UNO SCENARIO

Descrizione: Il paziente autenticato chiude uno scenario.

Attori: Il paziente autenticato.

Flusso principale: Il paziente autenticato può chiudere lo scenario in qualsiasi momento senza penalità o al termine dell'esercizio.

Postcondizioni Il sistema salva la performance del paziente nel caso l'esercizio sia stato completato o si sia raggiunta una situazione per cui non esiste più una mossa valida.

Questo caso d'uso rappresenta la possibilità per il paziente di abbandonare l'esercizio in qualsiasi momento.

5.2 Diagramma di sequenza

Il successo passo della progettazione è stato quello di comprendere a pieno la sequenza di azioni degli attori coinvolti nell'approccio alla riabilitazione cognitiva, lo schema qui presentato (Figura 5.2) è una semplificazione piuttosto efficace di come poi è stato implementato il sistema.

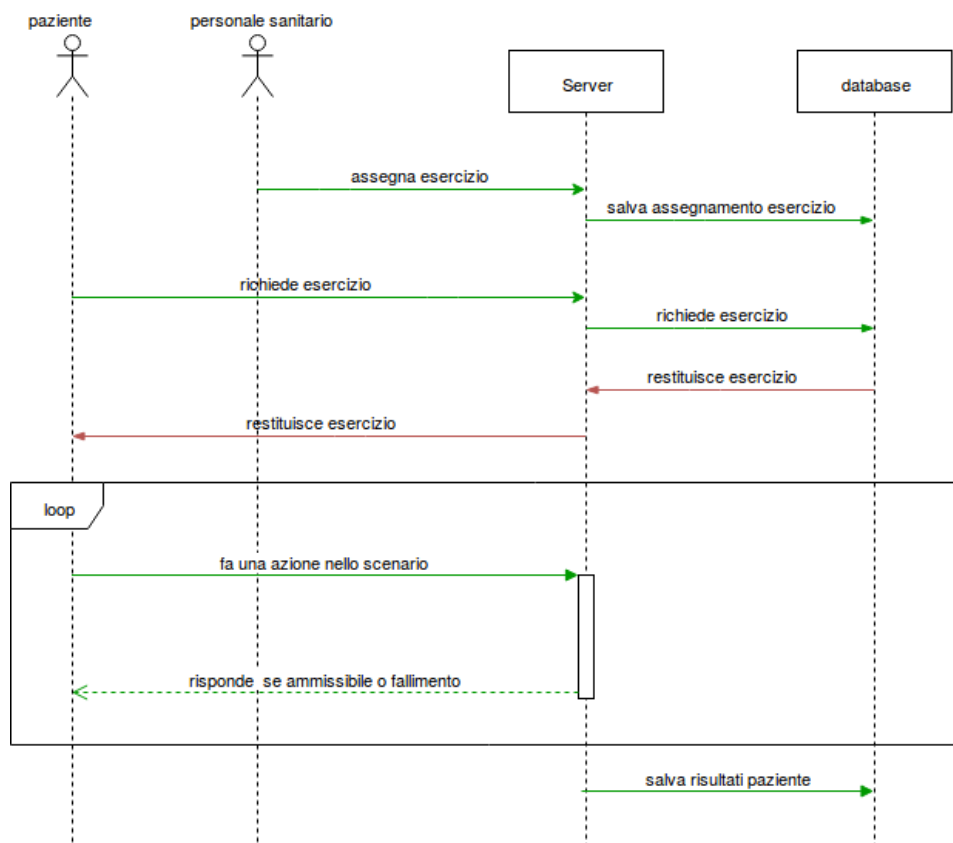


Figura 5.2: Il diagramma di sequenza della riabilitazione

Come è possibile vedere la richiesta di un nuovo esercizio da parte del paziente scatena nel server una serie di operazioni che coinvolgono la base di dati. La

base di dati viene interrogata nella ricerca di un esercizio adeguato al livello di difficoltà raggiunto dal paziente. L'esercizio viene poi inviato al browser, il paziente potrà quindi compiere le azioni per concludere con successo o con fallimento l'esercitazione al termine il sistema salverà i risultati ottenuti.

Completato l'esercizio al paziente sarà suggerito un esercizio successivo consono ai risultati ottenuti nell'esercizio precedente.

5.3 Strutturazione degli oggetti

Al fine di garantire una facile manutenzione del codice e poter costruire su di esso futuri aggiornamenti si è scelto di realizzare la documentazione relativa al progetto generando a priori il diagramma delle classi, ciò si è rivelato essenziale nelle successive fasi di sviluppo, disaccoppiando infatti le dipendenze specifiche e sfruttando alcuni design pattern ben noti in letteratura si è potuti procedere nello sviluppo di un software complesso ma efficiente [14].

La struttura del progetto lato server può essere agilmente descritta da un diagramma delle classi (Figura 5.3) che ne mette in luce punti di forza e principali limiti.

Il sistema può essere considerato un'entità separata dal resto del programma SM-Rehab in quanto è sufficiente importare la classe *PDDLProblemBuilder* per avere tutte le funzionalità del sistema. La classe principale infatti si preoccupa, data una serie di parametri in un certo *environment* di generare un nuovo problema. Il problema generato è un oggetto chiamato *PDDLproblem* dotato di una lista di *action* per indicare le azioni necessarie a completarlo, *belief* ovvero l'insieme di predicati atti a rappresentare lo stato del problema e *goals* ovvero gli obiettivi dell'esercizio. Il problema contiene anche un oggetto *json* che rappresenta la mappa dell'esercizio.

Tra le classi descritte si è applicato il *Builder pattern* due volte. Il *builder* è stato scelto perché le informazioni necessarie a creare un problema superano di gran lunga quelle necessarie alla descrizione dello stesso, ad esempio la topologia della mappa ha un numero piuttosto elevato di classi coinvolte che non sono necessarie alla rappresentazione della mappa al termine della creazione. Discorso analogo può essere fatto per il *builder* che si occupa della creazione dei *goal*, una volta generato il sotto problema si può eliminare il *builder* senza perdere informazioni importanti.

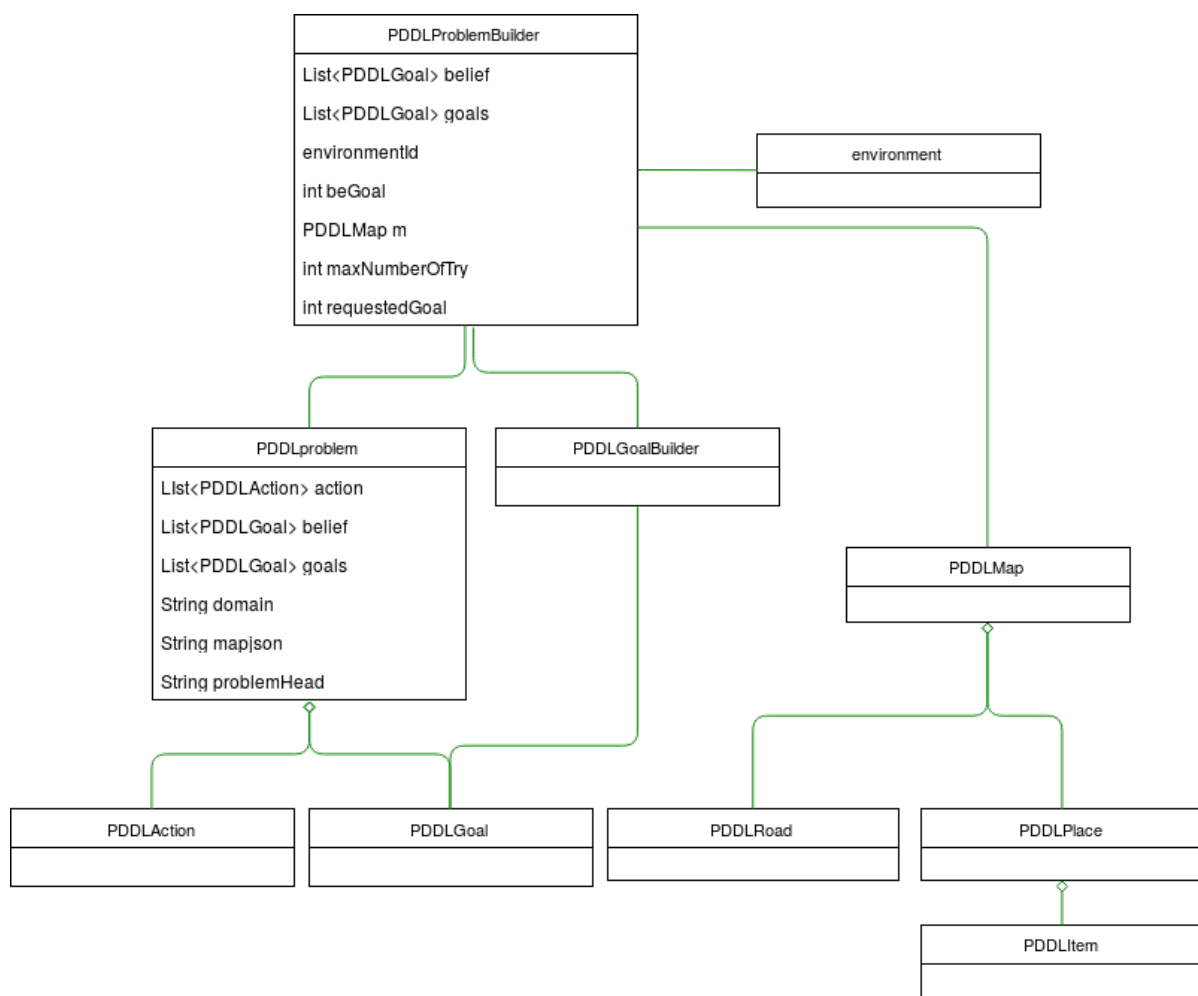


Figura 5.3: Diagramma delle classi del sistema

5.4 Base di Dati di MS-Rehab

Gli esercizi di riabilitazione delle funzioni esecutive sono stati inseriti in un sistema piuttosto complesso e già ricco di funzioni ed esercizi. Il diagramma presentato in figura 5.4 rappresenta lo schema delle entità presenti nel sistema e delle loro relazioni. In azzurro sono state segnate quelle entità correlate con i dati anagrafici, psicologici e sanitari del paziente, come il suo profilo, i test psicologici svolti e altri dati personali. In rosso si ha invece la parte più centrale del sistema dove stanno i dati dell'utente, i gruppi riabilitativi e le sessioni di esercizi assegnate. In viola gli esercizi stessi e i parametri atti a generarne dei nuovi, è importante notare come il sistema adotti per la maggior parte degli esercizi un approccio generativo, ovvero

non salva gli esercizi nel database ma solo i parametri necessari a configurare un nuovo esercizio. Infine in giallo vi è la parte riguardante lo storico degli esercizi svolti da ciascun paziente e la tabella che memorizza eventuali variazioni sulla difficoltà apportate dagli operatori.

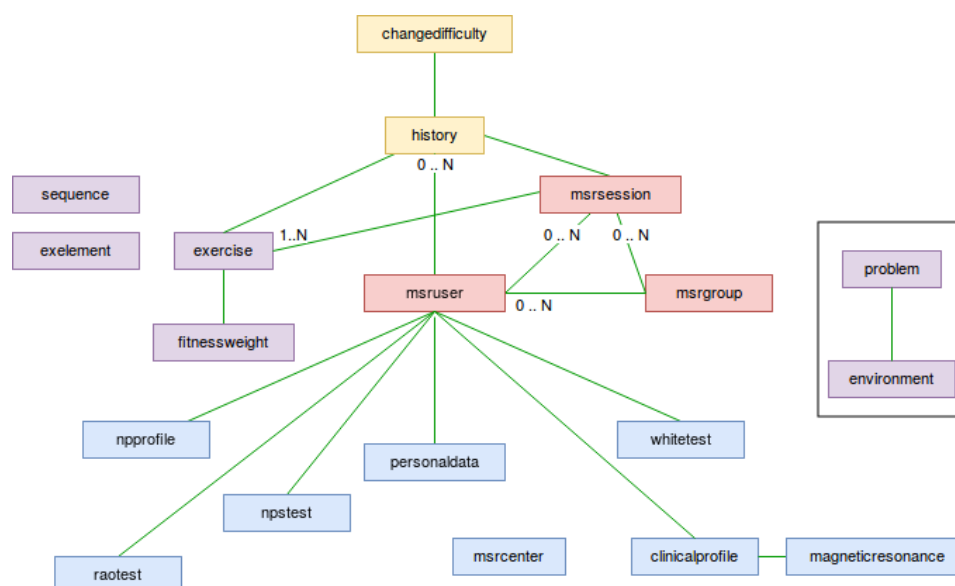


Figura 5.4: Diagramma completo delle Entità Relazioni del sistema, la parte evidenziata rappresenta le modifiche apportate descritte in Figura 5.5

Fatte queste doverose premesse data la complessità degli esercizi legati alle funzioni esecutive un approccio simile a quello degli altri esercizi non era possibile. È stato perciò necessario aggiungere due tabelle, di cui abbiamo il dettaglio in figura 5.5, tali tabelle sono atte a memorizzare gli esercizi generati e le loro caratteristiche principali. Ogni esercizio può essere facilmente espresso dalle sue componenti principali, la mappa salvata come json, la lista di goals e belief la sua soluzione e l'insieme delle entità coinvolte (*problemHead*). Quando al paziente viene assegnato un esercizio non viene scelto un esercizio specifico ma solo un livello di difficoltà, al sistema spetta il compito di scegliere l'esercizio data la difficoltà; per questo per ciascun problema viene memorizzata anche la relativa difficoltà, ovvero un parametro che lega la dimensione della mappa al numero di passi necessari per raggiungere la soluzione.

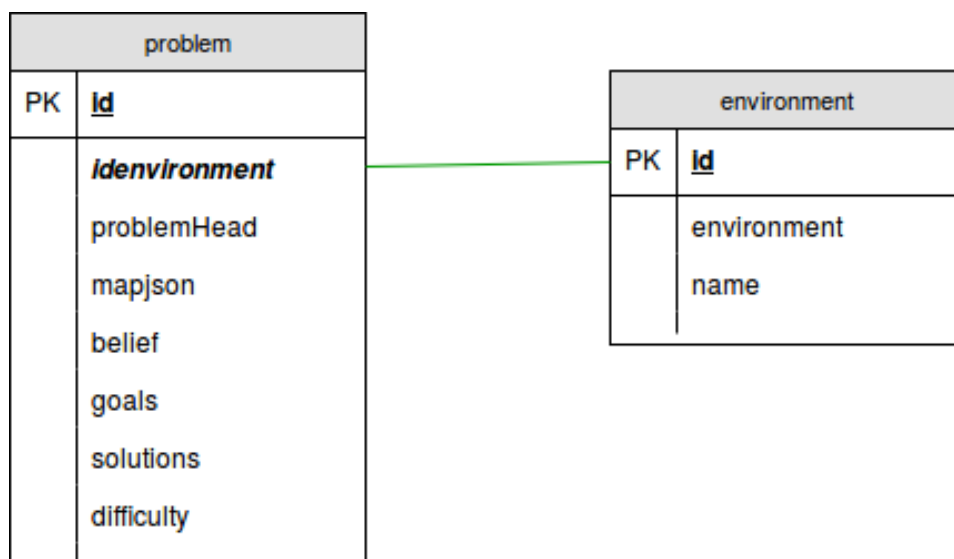


Figura 5.5: Dettaglio del diagramma Entità Relazioni del sistema

5.5 Grafica

Durante l'analisi dei modelli riabilitativi proposti si è individuato che le mappe proposte non rappresentavano situazioni reali né i luoghi rappresentavano modelli realistici degli edifici nella realizzazione si è perciò deciso di implementare i luoghi per mezzo di icone che potessero esprimere in modo chiaro e non ambiguo ciascun elemento presente in una mappa (Figura 5.6). Sono stati individuati perciò gli edifici principali da proporre nella generazione di nuovi esercizi gli edifici realizzati sono i seguenti: *casa, banca, piscina, supermercato, farmacia, gelateria, stazione e discarica*.

Essendo il sistema MR-Rehab una web app e volendo integrare questa nuova tipologia di esercizi è stato necessario sfruttare al meglio le differenti tecnologie necessarie a progettare un software con una così preponderante componente grafica.

Come è possibile vedere in figura 5.7 gli edifici vengono disposti secondo una griglia collegati da strade asfaltate con marciapiede (Figura 5.8). Il segnalino può essere mosso attraverso le strade trascinandolo con il dito.

Questa rappresenta una sfida ulteriore per il paziente che deve essere capace di seguire le strade senza poter semplicemente cliccare sulla destinazione. Ciascun edificio ha evidenziato in rosso il nome e gli orari di apertura e chiusura. Alcune strade di colore marrone o color asfalto perciò senza marciapiede possono essere attraversate rispettivamente solo a piedi o solo con un veicolo (Figura

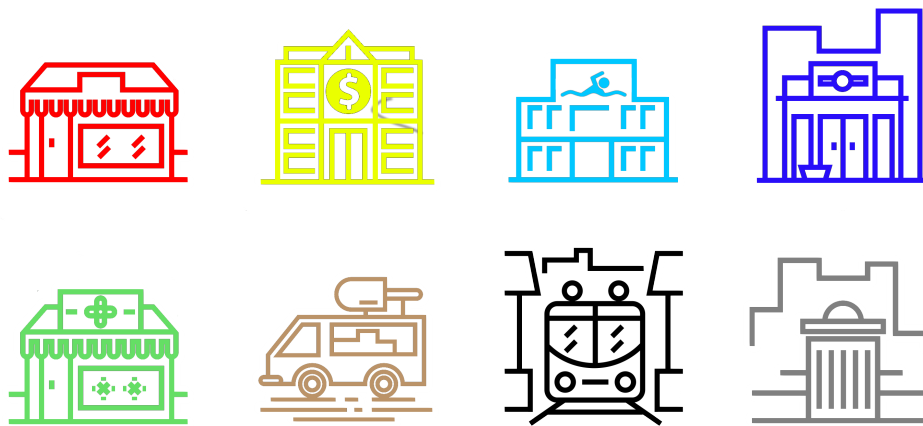


Figura 5.6: Esempio dei locali del sistema

5.8). Nel sistema è possibile esplicitamente aspettare, in alcuni luoghi, per far scorrere il tempo, quando il bottone è premuto al paziente viene richiesto di decidere fino a che ora aspettare. Ciascun edificio è dotato di una serie di oggetti, il paziente può raccogliere tali oggetti e posizionarli nell'inventario o depositare gli oggetti dell'inventario nel luogo in cui si trova. Sebbene la grafica sia minimale è stato necessario prendere una serie di accorgimenti per massimizzare l'usabilità del sistema, alcune scelte forzate come il trascinare il puntatore sullo schermo rappresentano vincoli all'usabilità del sistema per favorire il processo riabilitativo. Laddove possibile si è scelto il modo migliore per rendere semplice l'utilizzo del sistema fin dal primo accesso.

5.6 Descrizione del compito

Gli esercizi presi a modello e presentati nella sezione 2.4 hanno tutti liste di compiti da svolgere, ciascun elenco presenta in modo disordinato differenti richieste. Basandosi perciò sugli obiettivi e i vincoli individuati nel capitolo 4.1 si è cercato di implementare un generatore di esercizi che, data una mappa dotata di edifici, tessuto stradale e oggetti negli edifici, potesse creare un elenco di obiettivi con le stesse proprietà di *Mattina di impegni*. Si parla perciò di esercizi che coinvolgessero il paziente in un ragionamento più profondo sui vincoli del sistema. Se ad esempio al paziente venisse richiesto di: "prendere il gelato" sarà compito del paziente ricordarsi di rispettare gli orari di apertura e chiusura della gelateria e l'informazione che il gelato si trova effettivamente in gelateria. Questo

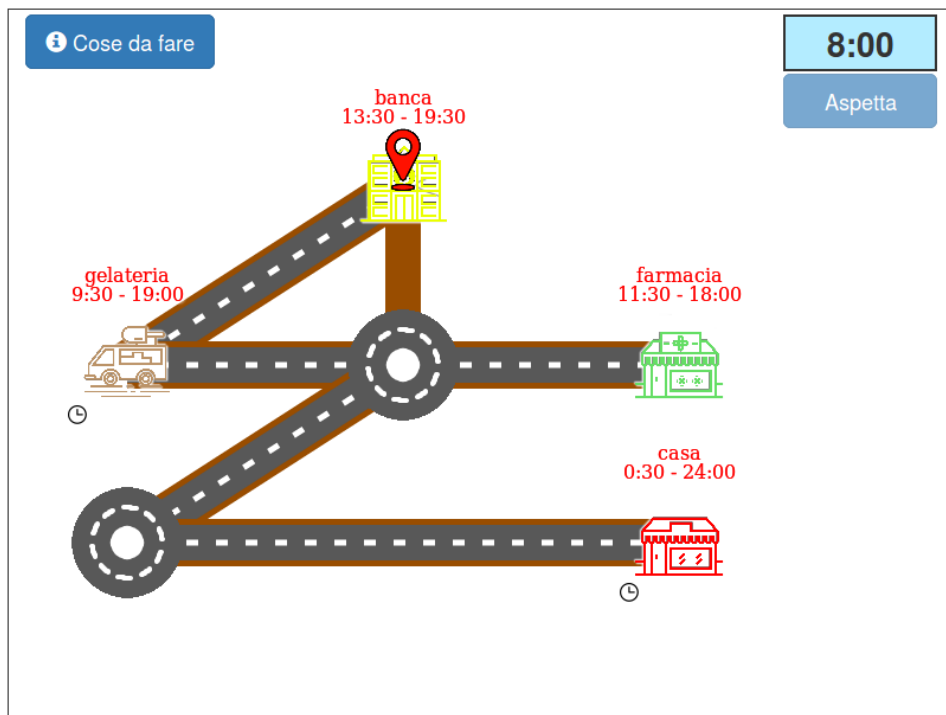


Figura 5.7: Un esempio di mappa generata dal sistema



Figura 5.8: Esempio di strade del sistema, sentiero pedonale, strada asfaltata, strada con marciapiede

che apparentemente risulta un compito molto semplice per una persona sana può rappresentare un ostacolo significativo per un paziente affetto da sclerosi multipla.

Ad esempio l'esercizio presentato in seguito ha una descrizione sommaria degli orari di apertura e chiusura dei negozi e successivamente una lista non ordinata di compiti da svolgere.

Vincoli:

1. La **discarica** è aperta dalle **12:30** alle **19:30**
2. La **piscina** è aperta dalle **9:30** alle **19:00**
3. La **farmacia** è aperta dalle **11:30** alle **17:30**

4. Il **supermercato** è aperto dalle **10:00** alle **18:30**

Obiettivi:

1. Fai in modo di essere al **supermercato** alle **15:30**
2. Prendi la **carne**
3. Come ultima cosa devi trovarti in **discarica**
4. Fai in modo di essere in **piscina** alle **16:30**

Come è possibile vedere l'obiettivo 2 impone una deduzione ovvero capire che la *carne* può essere acquistata al *supermercato*. Visto che il paziente ha il compito di passare dal supermercato alle 15.30 ed essendo il supermercato aperto a quell'ora potrà comodamente acquistare la carne in quella occasione, prima però dovrà passare dalla piscina. A fine esercizio dovrà invece tornare in discarica che è la posizione di partenza. Questo è sicuramente un punto che complica l'esercizio in quanto l'unico modo per concluderlo è proprio ripercorrere a ritroso tutti gli step fatti in precedenza (Figura 5.9).

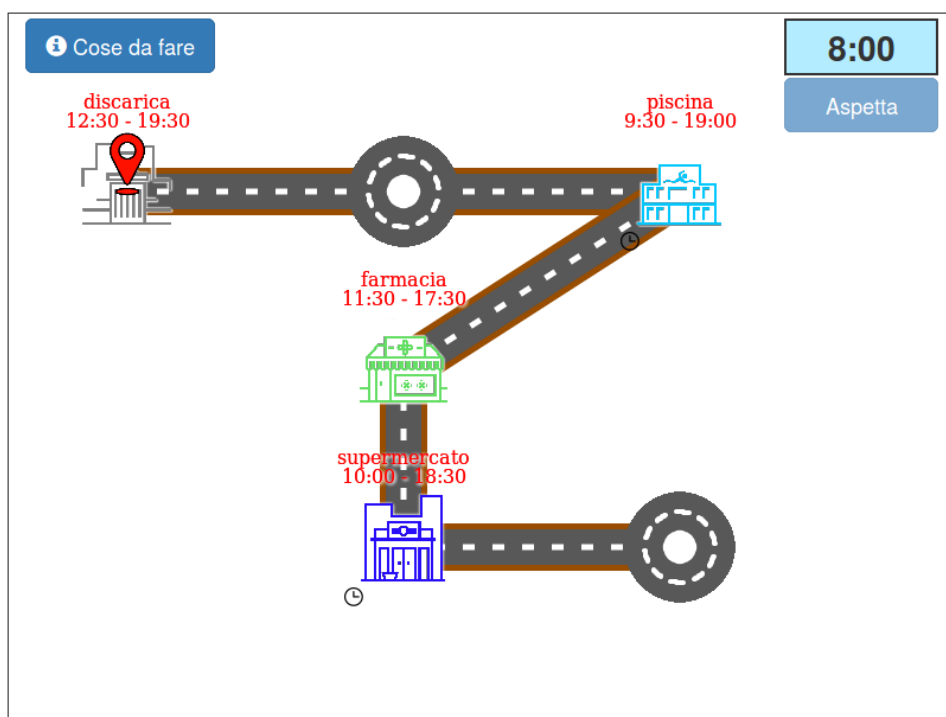


Figura 5.9: Un esempio di mappa generata dal sistema

5.7 Valutazione delle performance

Quando un paziente porta a termine un esercizio è opportuno assegnargli un punteggio sulla base delle performance raggiunte durante lo svolgimento. Purtroppo non è semplice decidere una metrica in uno scenario di pianificazione dove la soluzione è arbitrariamente complessa, fortunatamente i problemi generati in questi scenari sono per costruzione risolvibili in un numero finito di passi, tale numero è ottimo. Possono coesistere differenti soluzioni ma tutte isomorfe nel numero dei passi. Pertanto è possibile stabilire una gerarchia di punteggi sulla base delle differenti situazioni che si vanno a creare.

Il sistema è strutturato per valutare 4 componenti fondamentali, numero di risposte corrette, numero di risposte sbagliate, numero di risposte omesse, e tempo impiegato. Tali parametri vengono composti in una formula chiamata performance visibile nell'equazione 5.1.

$$Punteggio = \frac{pC * p1}{pTOT} + \frac{pM * p2}{pTOT} + \frac{pW * p3}{pTOT} + \frac{(maxT - pT) * p4}{maxT} \quad (5.1)$$

Tale formula pesa ciascuno dei parametri definiti pC , pM e pW ovvero risposte corrette, risposte omesse e risposte sbagliate e il tempo impiegato pT . Ciascuno di questi parametri è pesato in modo differente a seconda dell'esercizio, i pesi rappresentati da $p1$, $p2$, $p3$, $p4$ possono assumere valori positivi e negativi ma la loro somma da sempre 1. Tale sistema assicura che esercizi molto diversi possano essere valutati dalla medesima funzione in termini di performance ma ciascun esercizi mantiene le sue proprietà, un esercizio per il quale le omissioni rappresentano un grave problema avrà un $p2$ molto grande in valore assoluto.

Si è cercato di far corrispondere i passi fatti dall'utente nella soluzione a queste categorie. Se l'esercizio è completato con successo il numero di risposte corrette (pC) è il numero di mosse ottime per risolvere l'esercizio mentre il numero di risposte omesse (pM) rappresenta i passi in più richiesti dal paziente (Equazione 5.2). In questo esempio il malato ha scelto un cammino peggiore per trovare la soluzione che pertanto lo penalizza in termini di punteggio.

$$omesse = passi_fatti - passi_soluzione \quad (5.2)$$

Se l'esercizio è fallito e il paziente ha compiuto $passi_fatti$ allora si calcola la soluzione partendo da $passi_fatti - 1$ tale soluzione è compiuta da un dato numero di passi e può essere valutata dall'equazione 5.2 per calcolare quanti passi si discostano dal cammino migliore. Gli errori invece sono i passi necessari a raggiungere la soluzione da $passi_fatti - 1$ fino alla soluzione.

$$\text{errori} = \text{passi_soluzione_da_passi_fatti} - \text{passi_fatti} \quad (5.3)$$

Questo metodo per il calcolo delle performance descritto nell'equazione 5.1 è stato sviluppato da me durante il tirocinio presso il DESI. La formula sviluppata per essere il più generale possibile trova la sua applicazione in tutti gli altri esercizi del sistema, permette infatti di valutare con precisione l'abilità del paziente nel risolvere gli esercizi e rappresenta una buona metrica per seguire passo passo la loro riabilitazione. La formula è stata scritta in collaborazione con gli psicologi dell'ospedale Bellaria di Bologna prendendo in esame le componenti fondamentali per una corretta rappresentazione delle performance, il tempo infatti gioca un ruolo chiave nella patologia della sclerosi multipla e un miglioramento nei tempi di risposta rappresenta un grande miglioramento per la vita del paziente, discorso analogo per le risposte corrette, omesse per distrazione e sbagliate. Questa formula permette di disegnare grafici precisi sullo stato riabilitativo del paziente e monitorare nel corso del tempo le sue performance.

Capitolo 6

Test

Al fine di presentare un prodotto stabile e completamente funzionante, adatto ad un pubblico ampio e talvolta, soprattutto nel caso dei pazienti, non esperto, si è deciso di procedere con una serie di test al fine di verificare il corretto funzionamento e la bontà delle soluzioni proposte. Si è perciò deciso da un lato di presentare un test sull'efficienza dei generatori di problemi nonché di soluzioni per valutare che il sistema fosse in grado di fornire problemi sempre nuovi e stimolanti e non incontrasse bug di sorta durante la generazione, dall'altro lato un test di usabilità che potesse rappresentare un primo passo verso ulteriori test più approfonditi su pazienti. Il test così svolto ha una valenza preliminare e assicura il corretto funzionamento del software nonché la completa capacità da parte di utenti, non precedentemente istruiti, di fruire a pieno delle possibilità offerte dal programma e di risolvere con successo batterie di esercizi.

Nella valutazione si è tenuto conto oltre che al funzionamento del programma della soddisfazione individuale dei soggetti intervistati nello svolgere un esercizio, delle eventuali difficoltà incontrate dagli stessi a causa dell'interfaccia del programma.

6.1 Efficienza dei generatori

Il sistema è strutturato per poter progettare nuovi problemi al fine di avere una batteria di esercizi sempre varia da proporre ai pazienti. Ci si è quindi concentrati sulla distribuzione della difficoltà dei problemi in funzione dei suoi parametri base, ovvero il numero di goal. Si è perciò richiesto al sistema di generare 167 problemi con un numero di obiettivi compresi tra 2 e 6. La relazione tra numero di goal e difficoltà del problema non è lineare infatti ci sono problemi che richiedono una

soluzione molto complicata nonostante l'esiguo numero di goal. La difficoltà di un problema viene calcolata come segue:

$$Difficolta' = step_number - min_number_of_step + \frac{place_number}{3} \quad (6.1)$$

step_number: rappresenta il numero di azioni necessarie a risolvere l'esercizio.

min_number_of_step: nel sistema configurato a 4 rappresenta il minor numero di azioni per cui un problema è considerato facile. Tutti i problemi devono garantire almeno 4 azioni per poter essere risolti.

place_number: rappresenta il numero di locali presenti nella mappa, è un modificatore della formula in un range tra 1 (con 3 locali) fino a 4 (con 12 locali).

In generale perciò la distribuzione di probabilità con cui vengono generati gli esercizi è di tipo normale. In figura 6.1 viene mostrata la classica gaussiana con il picco su problemi con difficoltà 7.

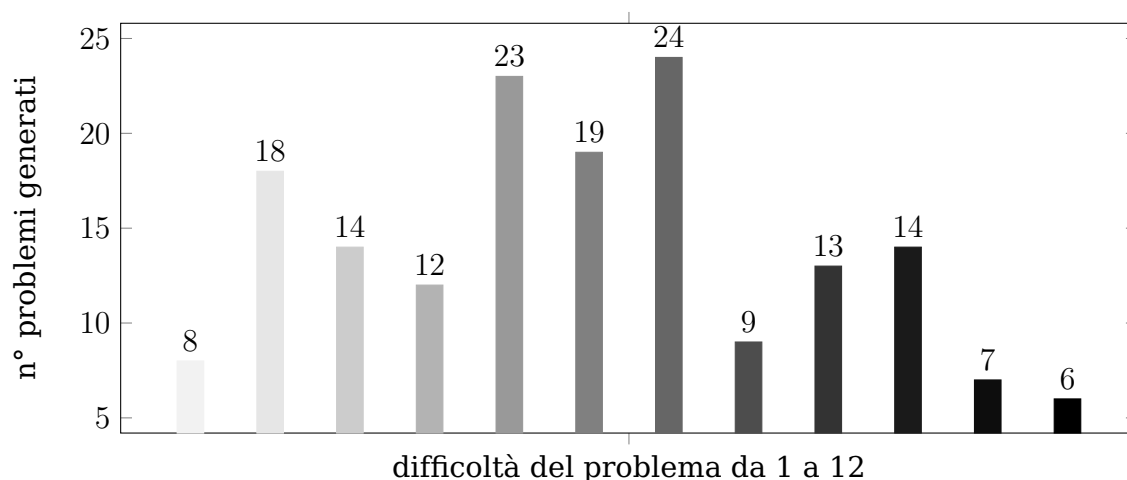


Figura 6.1: Distribuzione della difficoltà in 120 problemi simulati

Il sistema è stato testato su una macchina Linux 64 bit, Intel(R) Core(TM) i7-4790K CPU @ 4.00GHz, con 16 Gb di ram DD3 1600 MHz. Da questo test è stato possibile estrapolare il tempo medio per la generazione di esercizi a differenti livelli di difficoltà, il grafico presentato in figura 6.2 mostra una crescita lineare della tempo di esecuzione in funzione della difficoltà, ciò è ottimo in quanto in futuro sarà possibile generare esercizi con difficoltà anche maggiori in tempi contenuti.

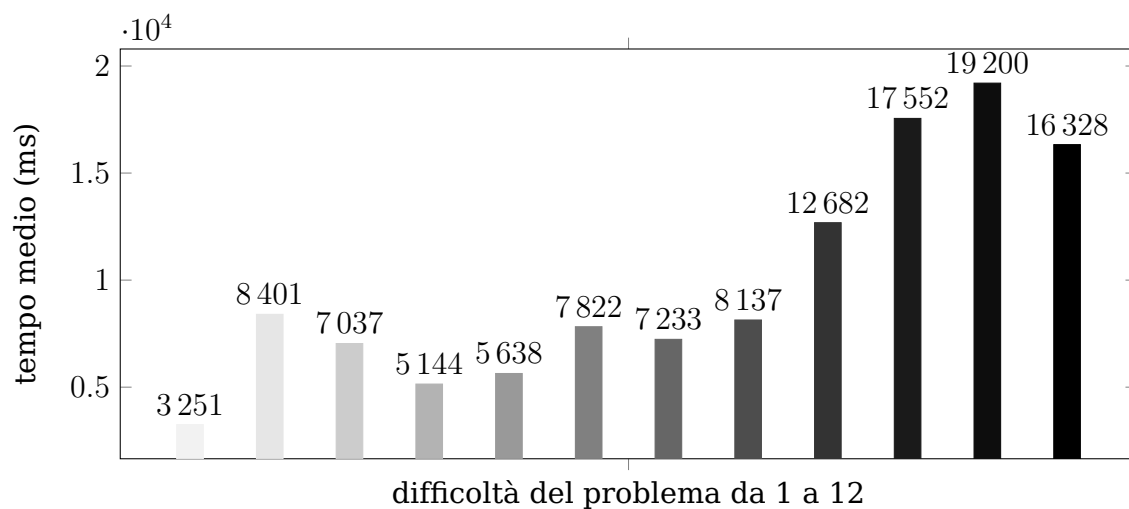


Figura 6.2: Distribuzione dei tempi (ms) di esecuzione di 120 problemi simulati rispetto alla loro difficoltà

6.2 Test Usabilità

Essendo il progetto focalizzato sullo sviluppo di una applicazione volta alla riabilitazione neurocognitiva delle funzioni esecutive in pazienti con sclerosi multipla è stato fondamentale svolgere alcuni test di usabilità su persone sane in modo da ridurre al minimo la difficoltà introdotta da un interfaccia digitale.

6.2.1 Protocollo di user testing

Si è scelto un protocollo di user testing chiamato *Discount usability testing* o più noto come *Guerrilla*, questa tipologia di test è adatta quando, per mancanza di tempo, si vuole sistemare un software senza scendere troppo nel dettaglio, è ottima come prima frontiera di risoluzione dei problemi a cui poi far seguire un test più approfondito seguito da uno psicologo.

Elenco dei task da testare

1. Comprensione del testo di un esercizio.
2. Avviare un esercizio.
3. Muovere il cursore in un nuovo edificio.

4. Raccogliere un oggetto.
5. Scambiare un oggetto con un altro oggetto.
6. Avere un suggerimento dal sistema circa quanto bisogna ancora svolgere.
7. Aspettare in un luogo per il tempo richiesto dall'esercizio.
8. Salire a bordo di un'automobile.
9. Individuare gli orari di apertura dei negozi.
10. Completare un esercizio con successo.
11. Interrompere un esercizio.
12. Nel complesso come ti è sembrata l'esperienza d'uso.

Numero di soggetti

Si è scelto di condurre l'esperimento su 7 persone scelte tra amici, colleghi e parenti.

Metodologia di testing

Per il testing si è scelta la metodologia dello *Thinking Aloud*, ovvero richiedere al soggetto di ragionare ad alta voce su ciò che sta svolgendo sul device, dando poi una valutazione sul singolo task ed infine sull'esperienza di utilizzo in generale.

Descrizione dei risultati attesi (esempio: metodologia Efficacia, Efficienza, Emozioni (EEE))

Gli errori riguardanti l'efficacia/efficienza durante i task possono essere:

Catastrofici: l'utente non conclude il task.

Gravi: l'utente conclude il task con rallentamenti significativi e/o con notevoli compromessi sulla qualità dell'output.

Cosmetici: l'utente identifica seccature nello svolgimento di uno o più task.

Per la valutazione delle emozioni di ciascun test si è chiesto al partecipante di rispondere con un punteggio da 1 (frustrante) a 5 (molto soddisfatto/intuitivo).

Scelta dei soggetti

Il target di utenza dell'applicazione è rappresentato da tutti i possibili pazienti affetti da sclerosi multipla che è una patologia che colpisce principalmente persone con più di 50 anni, perciò è legittimo assumere che il target scelto sia in qualche modo dipendente da questo parametro.

Organizzazione del test

1. Convocazione partecipanti al test in un ambiente consono.
2. Breve spiegazione del test con introduzione all'argomento.
3. Consegna del documento di usability test.
4. Consegna del tablet aperto sull'esercizio.
5. Test
6. Ringraziamenti.

6.2.2 Risultati del test sugli utenti

1. Comprensione del testo di un esercizio.

Soddisfazione media: 4

Note: Tutti gli utenti hanno compreso il testo dell'esercizio e sono riusciti a capire come svolgerlo in maniera corretta, sebbene molti al primo impatto si siano sentiti confusi dalla mancanza di ordine nei goal da conseguire (vincolo implementativo), sia se sia necessario o meno memorizzare gli obiettivi.

2. Avviare un esercizio.

Soddisfazione media: 5

Note: Tutti gli utenti hanno avviato l'esercizio senza difficoltà.

3. Muovere il cursore in un nuovo edificio.

Soddisfazione media: 4

Note: [Cosmetico] Un golfo piuttosto comune è stato il tentativo di toccare la destinazione per far muovere il cursore nella casella indicata, dopo questo primo tentativo la maggior parte degli utenti è riuscita a capire che per muovere il cursore occorre trascinarlo lungo lo schermo.

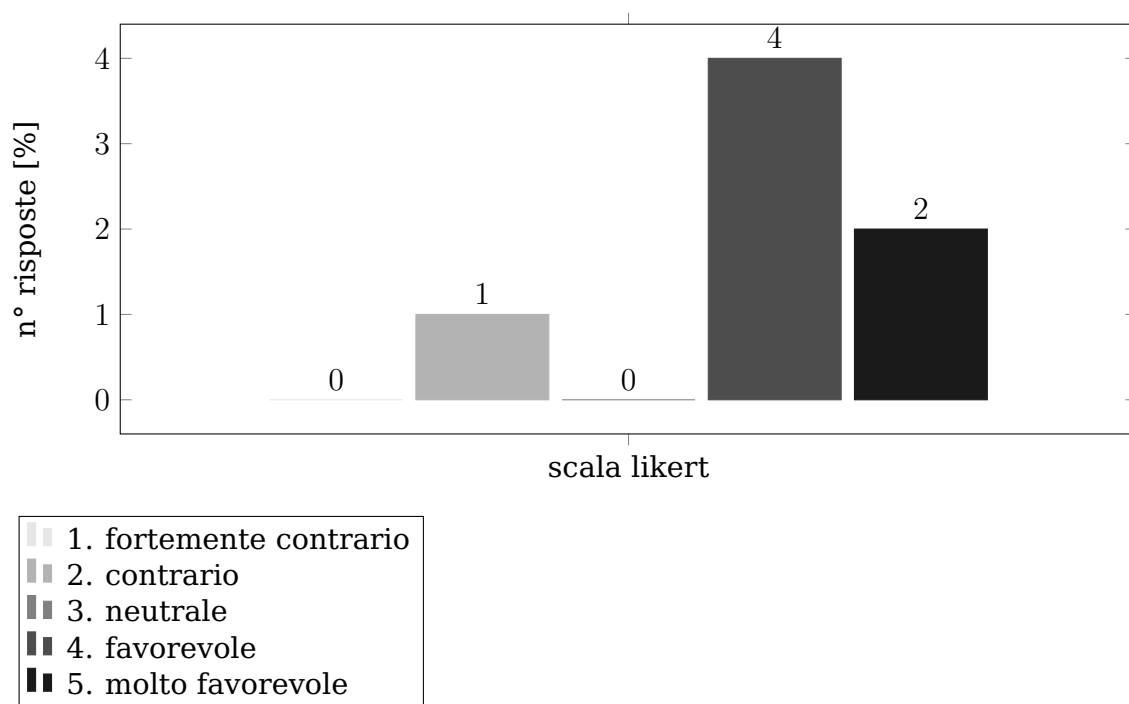


Figura 6.3: Discout usability testing domanda: "È stato semplice muovere il cursore in un nuovo edificio?" presentato secondo scala likert

4. Raccogliere un oggetto.

Soddisfazione media: 5

Note: Tutti gli utenti sono riusciti senza difficoltà a raccogliere un oggetto nello svolgimento dell'esercizio.

5. Scambiare un oggetto con un altro oggetto

Soddisfazione media: /

Note: Nell'esercizio proposto non era presente questa opzione.

6. Avere un suggerimento circa quanto bisogna ancora svolgere.

Soddisfazione media: 5

Note: Tutti gli utenti hanno compreso agilmente come rivedere l'elenco dei goal.

7. Aspettare in un luogo per il tempo richiesto dall'esercizio.

Soddisfazione media: 3

Note: Aspettare in un luogo si è rivelato uno dei compiti più frustranti e difficili compromettendo l'usabilità del software con errori sia [Cosmetici] che [Gravi], alcuni utenti hanno premuto sull'orologio e non sul bottone sottostante, altri invece hanno perso diversi secondi chiedendosi come far trascorrere il tempo.

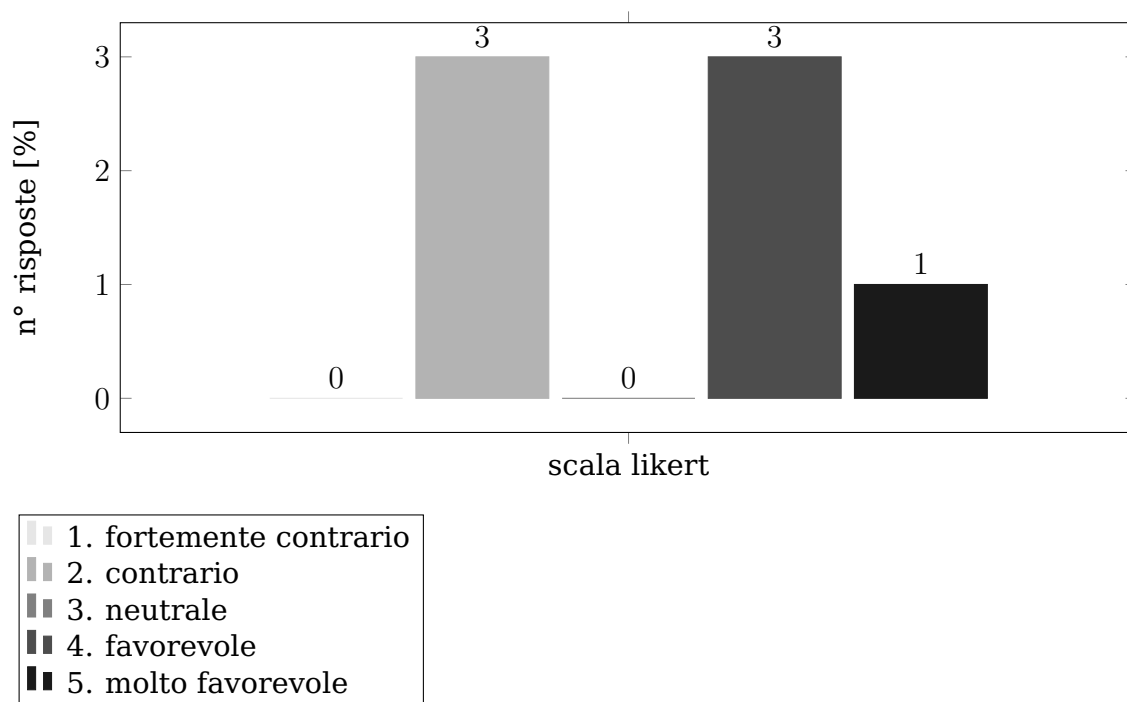


Figura 6.4: Discout usability testing domanda: "È stato semplice aspettare in un luogo?" presentato secondo scala likert

8. Salire a bordo di una automobile.

Soddisfazione media: /

Note: Nell'esercizio proposto non era presente questa opzione.

9. Individuare gli orari di apertura dei negozi.

Soddisfazione media: 5

Note: Tutti gli utenti hanno trovato estremamente semplice individuare gli orari di apertura.

10. Completare un esercizio con successo.

Soddisfazione media: 4

Note: Tutti gli utenti sono riusciti a completare l'esercizio con successo.

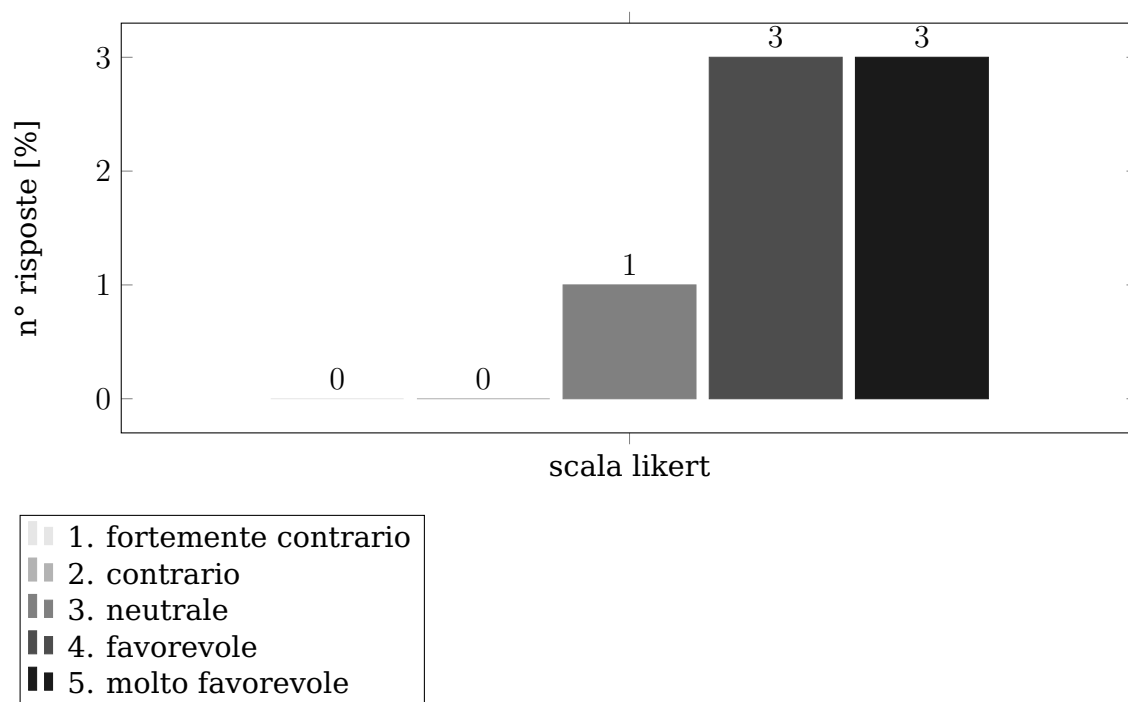


Figura 6.5: Discout usability testing domanda: "È stato semplice completare un esercizio con successo?" presentato secondo scala likert

11. Interrompere un esercizio.

Soddisfazione media: 5

Note: Tutti gli utenti sono riusciti ad interrompere un esercizio.

12. Nel complesso come ti è sembrata l'esperienza d'uso.

Soddisfazione media: 4

Note: Tutti gli utenti sono riusciti a sfruttare il software senza nessuna introduzione all'esercizio e senza nessuna spiegazione sul funzionamento dell'applicazione.

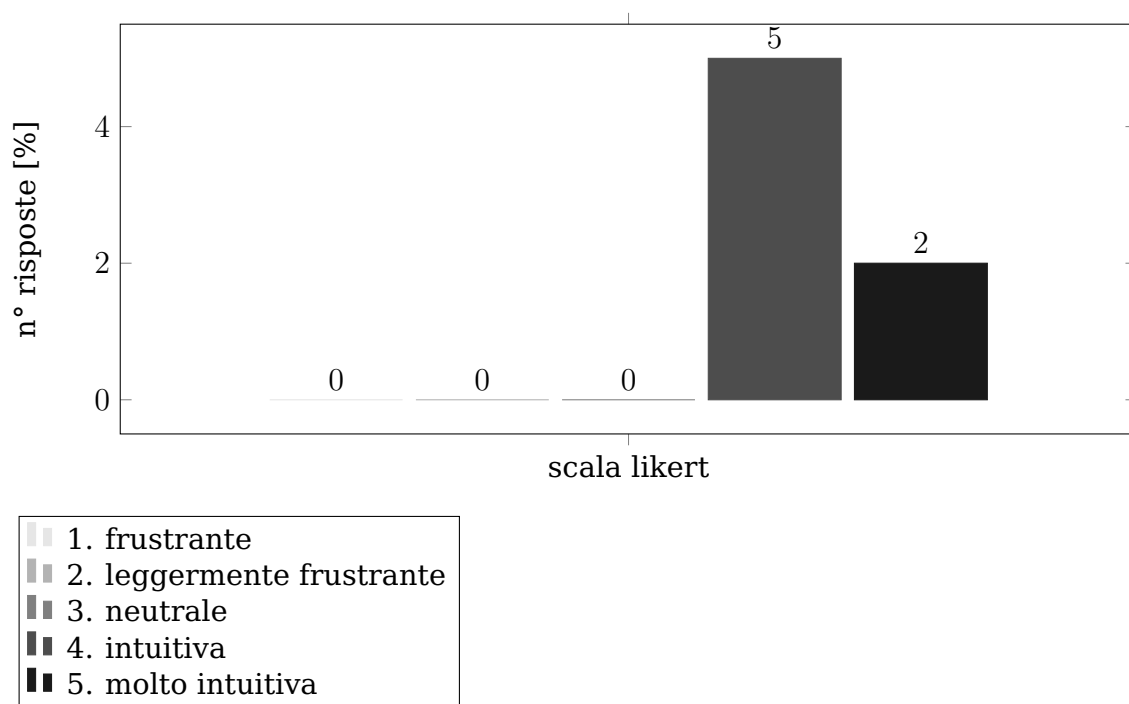


Figura 6.6: Discout usability testing domanda: "Nel complesso come ti è sembrata l'esperienza d'uso?" presentato secondo scala likert

6.2.3 Perfezionamenti successi alla valutazione

A seguito della valutazione del campione di testing si è pensato ad alcuni miglioramenti apportati conseguentemente:

Aspettare nei locali.

Si sono seguiti due approcci fondamentali alla risoluzione del problema dei tempi di attesa nei locali, da un lato si è ingrandito il box dell'orario e si è aggiunta l'icona dell'orologio per far capire meglio che l'oggetto indicato è il display di un orologio (Figura 6.9) dall'altro lato si è scelto di rendere cliccabile anche il display dell'orologio e non solo il bottone sottostante migliorando l'area totale attesa a ricevere il tocco dell'utente.

Inoltre si è deciso di aggiungere nella legenda fin dalla prima schermata una semplice spiegazione sul funzionamento del bottone *Aspetta*, il testo proposto è il seguente:

"Permette di far scorrere il tempo, il tempo passa solo quando si decide di aspettare."



Figura 6.7: Modifica apportata dopo i test di usabilità

Tempi di percorrenza a piedi e in macchina.

Questo aspetto non presente nella versione proposta agli utenti è stato intuito dalla maggior parte di loro durante lo svolgimento del esercizio, la mancanza è stata segnalata da molti utenti e si è deciso di intervenire aggiungendo questa informazione nella legenda (Figura 6.8).

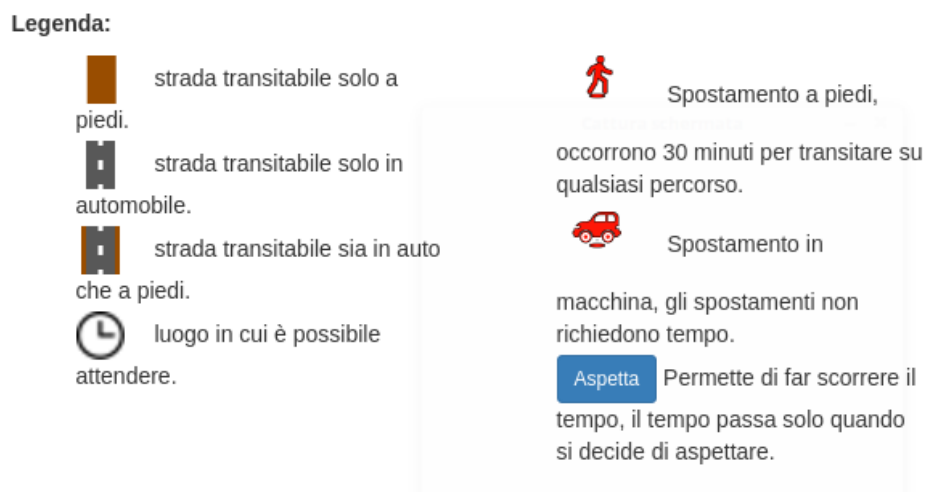


Figura 6.8: Modifica apportata dopo i test di usabilità alla legenda.

La posizione dell'utente non è chiara.

Diversi utenti hanno segnalato la necessità di rendere maggiormente chiara la pedina che identifica la posizione del utente sulla mappa. La scelta tipica delle mappe di *Google* per la sua facilità di identificazione e di utilizzo, è stata ampiamente respinta in fase di test. Si è perciò deciso di passare ad una figura stilizzata che potesse rappresentare in modo immediato l'utente.



Figura 6.9: Modifica apportata dopo i test di usabilità al cursore. Da sinistra vecchio modello utente, nuovo modello utente, modello utente in macchina.

Capitolo 7

Conclusioni

7.1 Conclusioni

In questa tesi si è sviluppato una componente essenziale di un software per la riabilitazione neurocognitiva di pazienti affetti da sclerosi multipla. L'applicazione web SM-Rehab mancava infatti di alcuni importanti esercizi di riabilitazione legati alle funzioni esecutive, che rappresentano un importante componente coinvolta direttamente in attività svolte giornalmente nella vita, e necessita quindi di strumenti riabilitativi adeguati. Il software così sviluppato può rappresentare un buon punto di partenza per sviluppare altri esercizi simili. Il programma ha raggiunto la maturità per poter essere somministrato ai pazienti essendosi dimostrato stabile e realmente utilizzabile, è stato infatti effettuato un test di usabilità su un gruppo di colleghi che ha messo in luce la semplicità di utilizzo del programma. L'esercizio realizzato verrà prossimamente utilizzato in una ricerca sperimentale che coinvolgerà tre centri Sclerosi Multipla in Italia: Fidenza, Parma e Firenze.

La soluzione proposta si è dimostrata all'altezza di scenari reali e con vincoli utili alla riabilitazione. Sebbene ci sia ancora qualche miglioramento da fare per ottimizzare ulteriormente il codice e estenderne le sue funzionalità. Il programma si presenta come un prodotto finito e pronto per la messa in produzione.

Sebbene manchino i dati di un test più esteso, i risultati preliminari mettono in luce la semplicità di utilizzo dell'esercizio realizzato e le grandi possibilità che questo può offrire in ambito medico. L'integrazione di questi esercizi riabilitativi nel più grande progetto SM-Rehab rappresentano sicuramente un buon punto di partenza per la riabilitazione delle funzioni esecutive.

7.2 Sviluppi futuri

Il progetto fin qui sviluppato ben si presta per un completo test di usabilità sui pazienti che sarà condotto contestualmente al prossimo esperimento realizzato per verificare l'efficacia di MS-Rehab e quindi anche di questo esercizio nella riabilitazione.

Come prima estensione del progetto si potrebbe considerare di generalizzare ulteriormente i vincoli in modo da poter gestire una maggiore quantità di scenari, comprendendo anche lo scenario dello *Zoo* e altri affini.

Per quanto riguarda lo sviluppo vero e proprio dell'applicazione sarebbe importante sviluppare altri scenari per stimolare su differenti ambiti i pazienti. Ad esempio lo scenario della *gita a Firenze*, ma anche altri come *parco dei divertimenti* o *ritirare i farmaci adeguati dentro ad una farmacia* dove i collegamenti rappresentano scaffalature. Queste modifiche potrebbero essere fatte mantenendo inalterato il codice solo creando nuovi *json* degli scenari e aggiungendo le opportune immagini.

Una modifica sicuramente importante potrebbe essere quella di migliorare ulteriormente la grafica degli edifici e degli oggetti, una maggiore uniformità di temi e una grafica personalizzata potrebbero dare un look più caratteristico e appetibile ad un paziente per continuare la sua sessione riabilitativa. Altra modifica sostanziale è l'aggiunta di un suggeritore che può essere sviluppato in due modalità distinte:

- Per far fronte ad eventuali momenti di difficoltà nella risoluzione di un esercizio da parte del paziente si potrebbe decidere di aggiungere un bottone o fissare un time-out che una volta attivo presenti la prossima mossa da svolgere verso la soluzione. Questo falserebbe l'autonomia del paziente ma potrebbe incentivarlo a continuare a giocare.
- Un secondo approccio potrebbe essere quello di raccontare l'intera soluzione al paziente in modo da invitarlo a ripercorrerla tramite i passi del programma, ciò andrebbe fatto solo le prime volte finché il paziente non è in grado di generare una soluzione da solo. Questo sarebbe in linea con gli studi fatti circa l'approccio riabilitativo alle funzioni esecutive tramite *generation effect* [8].

Bibliografia

- [1] Pddl - the planning domain definition language, 1997.
- [2] M. P. Amato, B. Goretti, R. G. Viterbo, E. Portaccio, C. Niccolai, B. Hakiki, P. Iaffaldano, and M. Trojano. Computer-assisted rehabilitation of attention in patients with multiple sclerosis: results of a randomized, double-blind trial. *Mult. Scler.*, 20(1):91–98, Jan 2014.
- [3] A. Ascherio and K. L. Munger. Environmental risk factors for multiple sclerosis. Part II: Noninfectious factors. *Ann. Neurol.*, 61(6):504–513, Jun 2007.
- [4] R. H. Benedict, D. Cookfair, R. Gavett, M. Gunther, F. Munschauer, N. Garg, and B. Weinstock-Guttman. Validity of the minimal assessment of cognitive function in multiple sclerosis (MACFIMS). *J Int Neuropsychol Soc*, 12(4):549–558, Jul 2006.
- [5] Gianni Biccone. *Un Sistema Intelligente per la Riabilitazione Cognitiva nella Sclerosi Multipla*. 2014.
- [6] S. Bonavita, R. Sacco, M. Della Corte, S. Esposito, M. Sparaco, A. d’Ambrosio, R. Docimo, A. Bisecco, L. Lavorgna, D. Corbo, S. Cirillo, A. Gallo, F. Esposito, and G. Tedeschi. Computer-aided cognitive rehabilitation improves cognitive performances and induces brain functional connectivity changes in relapsing remitting multiple sclerosis patients: an exploratory study. *J. Neurol.*, 262(1):91–100, Jan 2015.
- [7] A. Cerasa, M. C. Gioia, P. Valentino, R. Nistico, C. Chiriaco, D. Pirritano, F. Tomaiuolo, G. Mangone, M. Trotta, T. Talarico, G. Bilotti, and A. Quattro-ne. Computer-assisted cognitive rehabilitation of attention deficits for multiple sclerosis: a randomized trial with fMRI correlates. *Neurorehabil Neural Repair*, 27(4):284–295, May 2013.

- [8] N. D. Chiaravalloti and J. DeLuca. Cognitive impairment in multiple sclerosis. *Lancet Neurol*, 7(12):1139–1151, Dec 2008.
- [9] N. D. Chiaravalloti, H. Demaree, E. A. Gaudino, and J. DeLuca. Can the repetition effect maximize learning in multiple sclerosis? *Clin Rehabil*, 17(1):58–68, Feb 2003.
- [10] A. Compston and A. Coles. Multiple sclerosis. *Lancet*, 372(9648):1502–1517, Oct 2008.
- [11] L. De Giglio, F. De Luca, L. Prosperini, G. Borriello, V. Bianchi, P. Pantano, and C. Pozzilli. A low-cost cognitive rehabilitation with a commercial video game improves sustained attention and executive functions in multiple sclerosis: a pilot study. *Neurorehabil Neural Repair*, 29(5):453–461, Jun 2015.
- [12] D. A. Dymnt, G. C. Ebers, and A. D. Sadovnick. Genetics of multiple sclerosis. *Lancet Neurol*, 3(2):104–110, Feb 2004.
- [13] J. "Funke and T." Krüger. "Plan-A-Day": Konzeption eines modifizierbaren Instruments zur Führungskräfte-Auswahl sowie erste empirische Befunde. "J. Funke & A. Fritz (Eds.) *Neue Konzepte und Instrumente zur Planungsdiagnostik*", pages 97–120, 1995.
- [14] Erich Gamma, Richard Helm, Ralph Johnson, and John Vlissides. *Design Patterns: Elements of Reusable Object-oriented Software*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1995.
- [15] Mauro Gaspari, Floriano Zini, Debora Castellano, Federica Pinardi, and Sergio Stecchi. An advanced system to support cognitive rehabilitation in multiple sclerosis. In *Proceeding of IEEE RTSI 2017, 3° International Forum on Research and Technologies for Society and Industry*. IEEE Press & IEEE Xplore digital library, 2017.
- [16] J. Kroll, V. Karolis, P. J. Brittain, C. J. Tseng, S. Froudish-Walsh, R. M. Murray, and C. Nosarti. Real-Life Impact of Executive Function Impairments in Adults Who Were Born Very Preterm. "J Int Neuropsychol Soc", "23"("5"):"381–389", "May" "2017".
- [17] S. M. LaValle. *Planning Algorithms*. Cambridge University Press, Cambridge, U.K., 2006. Available at <http://planning.cs.uiuc.edu/>.
- [18] E. Làdavas. *La riabilitazione neuropsicologica*. 2012.

- [19] R. A. Marrie. Environmental risk factors in multiple sclerosis aetiology. *Lancet Neurol*, 3(12):709–718, Dec 2004.
- [20] U.S. Department of Health and Human Services. *Attention Deficit Hyperactivity Disorder (ADHD)*. Health & Outreach, revisited 2016 edition, 2009.
- [21] K. Rodewald, M. Rentrop, D. V. Holt, D. Roesch-Ely, M. Backenstrass, J. Funke, M. Weisbrod, and S. Kaiser. Planning and problem-solving training for patients with schizophrenia: a randomized controlled trial. *BMC Psychiatry*, 11:73, Apr 2011.
- [22] M.K.M. Sohlberg and C.A. Mateer. *Cognitive Rehabilitation: An Integrative Neuropsychological Approach*. Guilford Publications, 2001.
- [23] A. Tacchino, L. Pedulla, L. Bonzano, C. Vassallo, M. A. Battaglia, G. Mancardi, M. Bove, and G. Bricchetto. A New App for At-Home Cognitive Training: Description and Pilot Testing on Patients with Multiple Sclerosis. *JMIR Mhealth Uhealth*, 3(3):e85, Aug 2015.

Ringraziamenti

Il lavoro svolto in questa tesi e il relativo progetto è stato possibile grazie al sincero supporto di alcune persone che hanno speso tempo ed energie fornendo consigli e competenze.

Ringrazio perciò il mio relatore Chiar.mo Prof. **Mauro Gaspari** per avermi proposto questo stimolante progetto, per avermi sostenuto durante la stesura di questa tesi e per la grande cura che pone nel suo lavoro di docente universitario.

Ringrazio il mio correlatore il Dott. **Floriano Zini** per avermi introdotto a MS-Rehab, un progetto lungo e complesso, e avermi condotto passo passo finché non sono stato in grado di apportarvi modifiche in modo autonomo.

Ringrazio la collega **Elena Maria Bressan** per aver individuato tanti possibili errori nel programma prima che venisse rilasciato, per lo zelo adottato in questa meticolosa ricerca che ha permesso di sistemare enormemente il lavoro svolto.

Un ringraziamento speciale va ai **miei genitori** che mi hanno sostenuto in questi anni di studio permettendomi di fare ciò che davvero mi piace.

Infine ringrazio **Margherita** per essermi stata vicina in questi mesi impegnativi e ringrazio **i miei amici** che hanno deciso di esserci anche quando non era scontato.