Alma Mater Studiorum · Università di Bologna

SCUOLA DI SCIENZE

Corso di Laurea in Informatica Magistrale

Weekend in Rome
A Cognitive Training Exercise based on Planning

Relatore:
Chiar.mo Prof.
Mauro Gaspari

Presentata da:
Margherita Donnici

Correlatore:
Dott.
Floriano Zini

Sessione I
Anno Accademico 2017-2018

Alla mia famiglia

...

Sommario

La sclerosi multipla è una delle malattie più diffuse del sistema nervoso centrale: nel mondo si contano circa 2,5 milioni di persone con SM, di cui 600.000 in Europa e oltre 118.000 in Italia. Nello specifico, è una malattia neurodegenerativa demielinizzante: questo vuol dire che si verificano un danno e una perdita di mielina, una sostanza costituita prevalentemente da lipidi e proteine che riveste esternamente gli assoni dei neuroni. La mielina consente la corretta conduzione degli impulsi nervosi, amplificandone la velocità di trasmissione. Nella sclerosi multipla, le difese immunitarie del paziente attaccano e danneggiano questa guaina. Quando ciò accade, gli assoni non sono più in grado di trasmettere efficacemente i segnali. Questo produce i vari sintomi caratteristici della SM, che possono variare notevolmente e includono disturbi visivi, della sensibilità, della coordinazione, intestinali e vescicali. Inoltre, un certo livello di deficit cognitivo si manifesta nel 45-60% delle persone con SM. Questi disturbi cognitivi possono apparire negli stadi iniziali della malattia. I domini cognitivi danneggiati nella SM variano da paziente a paziente, tuttavia un *pattern* caratteristico può essere definito: in genere si tratta di disturbi dell'attenzione, del ragionamento, delle funzioni esecutive, della memoria e della percezione visuospaziale mentre il linguaggio è solo raramente compromesso. Benchè non esista nessuna cura farmacologica per la SM, sono tuttavia disponibili numerosi trattamenti che modificano il decorso della malattia e rendono possibile gestire ed alleviare i sintomi della malattia, al fine di ottenere un miglioramento della qualità di vita delle persone con SM.

La terapia di riabilitazione cognitiva, l'esercizio, e i programmi di educazione sono degli interventi psicosociali promettenti per gestire e diminuire i disturbi cognitivi. Nello specifico, la terapia di riabilitazione cognitiva è un termine usato per indicare i tratta-

menti che mirano a gestire questi disturbi. Recentemente, l'utilizzo di strumenti automatizzati per il training cognitivo nella SM (piuttosto che il tradizionale approccio carta e penna) è in continua crescita. Tuttavia, attualmente sul mercato sono disponibili solo strumenti *general-purpose*, non progettati su misura per malattie specifiche. Questa è stata la principale motivazione per lo sviluppo di MS-Rehab: un sistema avanzato capace di essere integrato nelle varie fasi del processo di riabilitazione cognitiva per la sclerosi multipla. MS-Rehab è frutto di una collaborazione tra il gruppo di ricerca del Prof. Mauro Gaspari del Dipartimento di Informatica - Scienza e Ingegneria ed i neurologi e psicologi dell'ospedale Bellaria di Bologna guidati dal Dott. Sergio Stecchi.

Lo scopo di questa tesi è stato l'implementazione di un esercizio per la riabilitazione delle funzioni esecutive da integrare nel sistema MS-Rehab. Con il termine *funzioni esecutive* viene denotato l'insieme dei processi mentali finalizzati all'elaborazione di schemi cognitivo-comportamentali adattivi in risposta a condizioni ambientali nuove e impegnative. Questi sono considerati componenti fondamentali della cognizione umana. Difatti, degli studi hanno dimostrato che dei deficit nelle funzioni esecutive possono portare a problemi sociali e comportamentali. La pianificazione è una delle abilità principali connesse alle funzioni esecutive; ed un requisito per numerosi *task* cognitivi e motori. Nello specifico, può essere definita come la capacità di organizzare comportamenti cognitivi nel tempo e nello spazio. Questa capacità è necessaria in particolare nelle situazioni in cui un obiettivo deve essere raggiunto tramite una serie di *step* intermedi, dove un singolo *step* non per forza porta direttamente all'obiettivo. La capacità di pianificare è precisamente l'abilità che viene allenata nell'esercizio realizzato in questa tesi, denominato "Weekend a Roma". Lo scenario che viene simulato è diverso da qualunque altro scenario presente nei diversi software di riabilitazione sul mercato ad oggi, per cui è stato necessario progettarlo partendo da zero con l'aiuto di psicologi ed esperti nella riabilitazione per assicurare la sua validità clinica. In questo esercizio, il paziente deve pianificare una vacanza di due giorni nella capitale italiana. Oltre a dover prenotare i treni e l'albergo, gli viene fornita una lista di *task* da compiere (quali posti da visitare o eventi a cui partecipare), superando tutte le difficoltà tipiche del panificare un viaggio nella vita di tutti i giorni (prenotazioni, orari degli autobus, orari di apertura...). Difatti, durante la progettazione, una grande importanza è stata data al rendere l'esercizio il più

realistico possibile, in modo che possa avere un'alta validità ecologica e per massimizzare l'impatto positivo sulla qualità della vita del paziente. Questo esecizio si basa sulla tecnica della pianificazione automatica, una branca dell'intelligenza artificiale collegata alla teoria decisionale che consiste nel concepire in maniera dinamica una certa sequenza di azioni che fan si che un certo obiettivo, non inizialmente verificato, venga raggiunto. Successivamente, l'esercizio è stato integrato nel sistema MS-Rehab incorporando diversi livelli di difficoltà auto-adattivi.

Introduction

Multiple sclerosis (MS) is one of the most common disease of the central nervous system, affecting about 2 1/2 million people worldwide. Specifically, it is an inflammatory demyelinating disease: this means it is caused by damage to the myelin, a lipid-rich substance that surrounds the axon of some nerve cells, and allows a nerve to transmit its impulses rapidly. Thus, the loss of myelin in MS is accompanied by a disruption in the ability of the nerves to conduct electrical impulses to and from the brain. This produces the various symptoms of MS, which vary widely and include deficits in movement, sensation, visual and bodily functions. In particular, a lot of researches indicate that 45 to 60% of patients are cognitively impaired [1]. Cognitive dysfunction could appear in the earliest stages of the disease as the first symptoms of MS [2]. The cognitive domains impaired in MS seem to have an inter-patient variability, but a characteristic pattern may be defined: memory, information processing efficiency, executive functioning, attention, processing speed, are the most commonly compromised functions. While there is no drug that can cure MS, treatments are now available which can modify the course of the disease and many of the symptoms of MS can be successfully managed and treated.

Cognitive rehabilitation therapy, exercise, and education programs are promising psychosocial interventions to improve coping and lessen cognitive symptoms. Specifically, cognitive rehabilitation therapy is a term used to describe treatments that address these cognitive problems. Recently, the use of computerized tools for cognitive training in MS has been increasing, as opposed to the traditional pen-and-paper approach. However, only general purpose tools which are not tailored for a specific disease are available in the current market. This was the main motivation behind the development of MS-Rehab: an advanced system able to integrate the various phases of the MS cognitive rehabilitation process. MS-Rehab was developed by Prof. Mauro Gaspari's research group of the Department of Computer Science and Engineering of the University of Bologna, in tight cooperation with the psychogists of the Bellaria Hospital in Bologna, lead by Dott. Sergio Stecchi.

The aim of this thesis is to implement a cognitive rehabilitation exercise for the rehabilitation of executive functions to integrate into the MS-Rehab system. *Executive function* is an umbrella term used to denote the set of higher-order processes widely accepted as fundamental components of human cognition; indeed, studies have shown that executive function deficits may lead to social and behavioral problems. Planning is one of the main skills related to executive functions, and a requirement of many cognitive and motor tasks. Specifically, it may be defined as the ability to organize cognitive behaviour in time and space and is necessary in situations where a goal must be achieved through a series of intermediate steps each of which does not necessarily lead directly towards that goal. The ability to plan is precisely the skill that the exercise realized in this thesis, named "Weekend in Rome", aims to train. The scenario which is simulated is different from any other present in rehabilitation software and had to be designed from scratch, involving a number of meetings with psychologists and rehabilitation experts to ensure its clinical validity. In this exercise, the patient has to plan a two-day vacation in the italian capital. In addition to making train and hotel reservations, he is also given a list of tasks (such as locations to visit or events to attend) he must accomplish, navigating all the difficulties which are typical of planning a trip in real life (reservations, bus schedules, opening hours...). Indeed, the exercise was carefully designed to be as realistic as possible, thus having a high ecological validity and to maximize the positive impact on the patient's quality of life. The exercise was built using automated planning, a branch of artificial intelligence related to decision theory, which involves devising a plan, described as a sequence of actions, in order to achieve a given goal. Subsequently, it was also integrated into the MS-Rehab system incorporating different auto-adaptive levels of difficulty.

The thesis is organized as follows:

Chapter 1: Background In this chapter, a brief introduction on executive functions and cognitive rehabilitation is given, followed by a survey on the current state of the art concerning Cognitive Rehabilitation tools and a presentation of the MS-Rehab system.

Chapter 2: Automated Planning In this chapter, a brief introduction on classical planning and the PDDL language is given.

Chapter 3: Weekend in Rome In this chapter, the design and implementation of the Weekend in Rome exercise will be described as well as its integration in the MS-Rehab system and a comparison with other existing exercises.

Contents

Chapter 1

Background

1.1 Executive Functions

Executive function is an umbrella term used to denote the set of higher-order processes (such as inhibitory control, working memory, and attentional flexibility) that govern complex reasoning, goal-oriented action and adaptive responses to novel, complex, or ambiguous situations [6]. These abilities include the maintenance and manipulation of information, temporal organization, set shifting, self-monitoring, concept formation, verbal fluency, inhibition, motivation, organization, and planning. Widely accepted as fundamental components of human cognition, studies have shown that executive function deficits may lead to social and behavioral problems, as well as academic underachievement[19]: people who have difficulty inhibiting themselves, remembering things, planning, problem solving, and being flexible will present major deficiencies in social, academic, and vocational functioning. In this chapter, we will give a brief introduction on executive functions and the state of the art for their rehabilitation, then present the MS-Rehab platform, an advanced system to support cognitive rehabilitation in multiple sclerosis developed at the University of Bologna.

1.1.1 Executive Functions

As stated previously, the term *executive function* covers a wide range of concepts: most researchers, in fact, agree that it is an exceptionally broad term, and there is little consensus on a precise definition [10]. Nevertheless, it is indisputable that the term refers to abilities which are critical for functioning in everyday life: for example, executive functions help you manage life tasks such as planning a trip, your day, a research project or even something as basic as preparing a meal. Metaphorically speaking, executive functions could be considered the brain's chief executive officer, the "conductor" of all cognitive skills. In fact, some abilities encompassed in executive functioning involve other cognitive domains as well, such as memory and attention.

The following specific abilities are covered under the umbrella term of executive functioning [5]:

- **Planning** - the ability to create steps to reach a goal and to make decisions about what to focus on.
- **Inhibition** - the ability to stop one's own behavior at the appropriate time, including stopping actions and thoughts.
- **Working memory** - the ability to hold information in mind and use it to complete a task.
- **Shift** - the ability to move freely from one situation to another and to think flexibly in order to respond appropriately to the situation.
- **Selective attention** - the ability of focusing on a particular task for a certain period of time, tuning out unimportant details and focusing on what really matters.
- **Self-Monitoring** - the ability to monitor one's own performance and to measure it against some standard of what is needed or expected.
- **Emotional Control** - the ability to modulate emotional responses by bringing rational thought to bear on feelings.
- **Task initiation** - the ability to recognize when it is the right time to begin a task or activity and to independently generate ideas, responses, or problem-solving strategies.

Executive functions gradually develop and change across the lifespan of an individual and can be improved at any time over the course of a person's life [9].

In the particular, the aim of this thesis is the design and implementation a cognitive rehabilitation exercise for the planning executive function. Specifically, planning may be defined as the ability to organize cognitive behaviour in time and space and is necessary in situations where a goal must be achieved through a series of intermediate steps, each of which does not necessarily lead directly towards the specified goal [22]. Simply put, planning is the ability to think about the future and mentally anticipate the right way to carry-out a task: this means to choose the necessary actions to reach a specific goal, decide the right order, assign each task to the proper cognitive resources, and establish a plan of action. In order to plan efficiently, one needs the necessary information and the ability to mentally establish an adequate synthesis of all the data.

1.1.2 Cognitive Rehabilitation

The ability to plan depends on elements like brain plasticity or neuroplasticity, myelinization, or the ability to establish new paths and synaptic connections. An absence or deficiency in this ability is a typical symptom of frontal lobe disorders, especially disorders that affect the prefrontal dorsolateral area. Any task that requires planning, organization, memorization, time management, and flexible thinking will be particularly challenging for subjects who have deficient planning abilities. Having significant difficulties in carrying out these kind of tasks is known as dysexecutive syndrome (DES) [7]. In everyday life, this means that a person's ability to care for himself, complete tasks, keep appointments and interact with people appropriately may be compromised. Without treatment, this may have devastating long-term effects on the ability to succeed at home, work or school.

Cognitive rehabilitation therapy (CRT) is a broad term used to describe treatments that address these cognitive problems. The Institute of Medicine's 2011 report defines CRT as cognitive rehabilitation attempts to enhance functioning and independence in patients with cognitive impairments as a result of brain damage or disease [8]. Pharmacological treatments have also been shown to be capable of altering cognitive functions and brain activations in healthy volunteers and patients [12].

In the following paragraphs, we will give a few example of cognitive exercises designed to test, restore or strengthen underlying cognitive functions, in particular planning, with a high ecological validity. Exercises with a high ecological validity are exercises which train the patient on typical tasks required in day-to-day life, thus giving rehabilitation a positive impact on the patient's quality of life.

Zoo Map In 1996, Wilson, Alderman, Burgess, Emslie, and Evans developed the Behavioral Assessment of the Dysexecutive Syndrome (BADS), an ecologically valid test battery designed specifically for the assessment of planning and organizational capacity for situations one could encounter in everyday life. Up until then, conventional executive function tests lacked ecological validity, resulting in a discrepancy between test performance and functioning in everyday life [15]. The entire battery contains six subtests, including the Zoo Map Test, where the participants are instructed to plan their route through a map of the zoo, visiting a selection of locations while actively disregarding others and obeying certain rules: for example, the order of the locations to visit, or crossing certain paths only once. This test is particularly useful to assess the specific executive deficit in order to carry out the appropriate rehabilitation therapy.

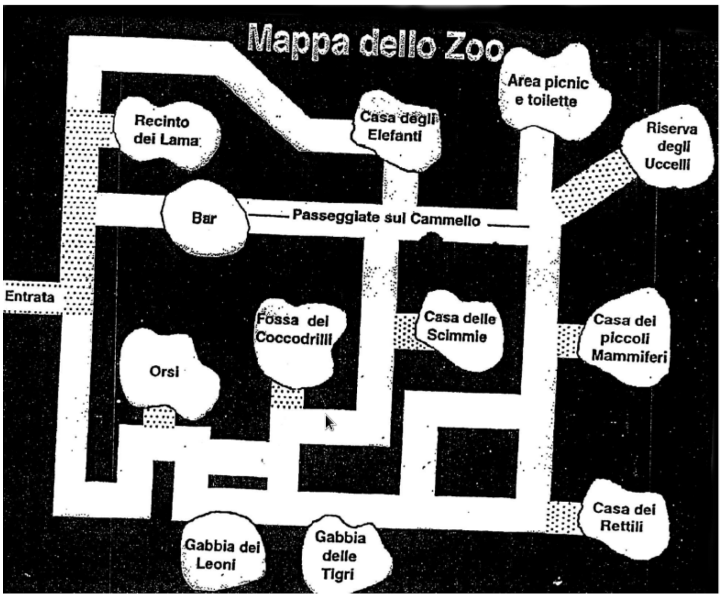

Figure 1.1: Example of a *Zoo Map* exercise

Plan-A-Day Another similar exercise is Plan-A-Day [11], proposed by Funke & Krüger in 1995. In this exercise, the patient has to schedule a list of tasks to complete during the day (for example, picking up his daughter from the swimming pool or buying groceries) while considering various constraints about when, where, and for what duration the activities have to be carried out.

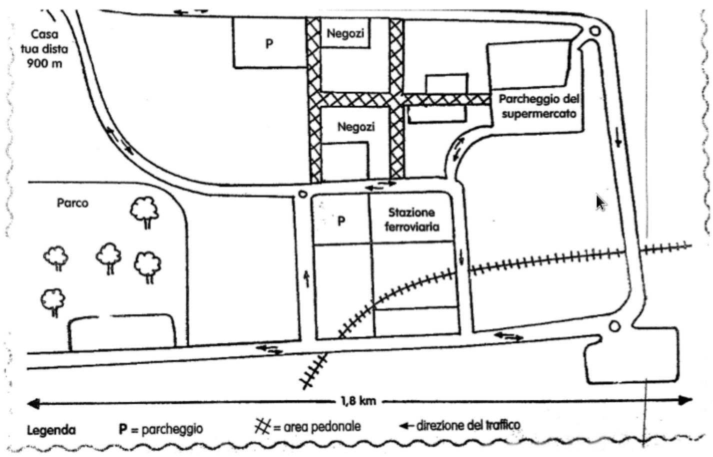

Figure 1.2: Example of a *Plan-A-Day* exercise

Map of Florence Another similar exercise used at the Bellaria Hospital in Bologna in which the patient has to visit a series of monuments in the city of Florence, meeting various constraints.

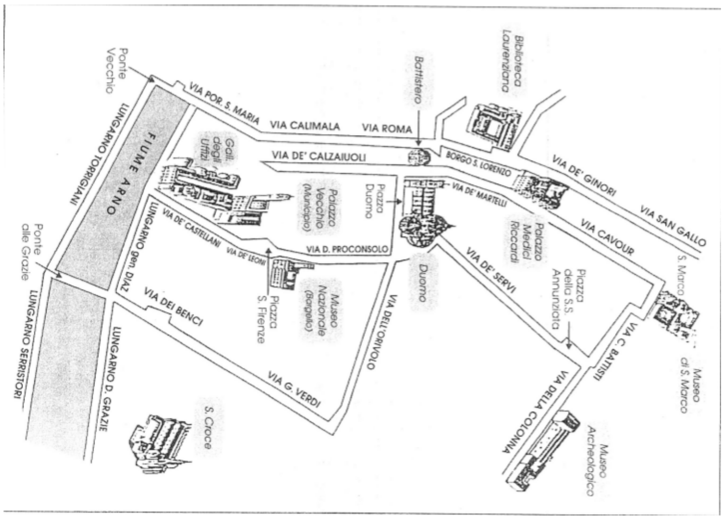

Figure 1.3: Example of a *Map of Florence* exercise

Traditionally, these exercises are carried out with pen and paper: the patient is given a copy of the map with the list of tasks to complete, then communicates his intentions by voice to the operator or traces his path on the map.

1.1.3 State of the art for rehabilitation

Currently, the most used approach for cognitive rehabilitation is the traditional one: the exercises are carried out with pen and paper, in the hospital, with the help of qualified staff such as psychologists and speech therapists. While this type of approach has the advantage of giving the operator complete control on the patient and his rehabilitation, it has the drawback of being quite demanding in terms of resources. First of all, it requires the presence of specialized staff dedicated to the single patient or group of patients. Moreover, continuous intervention is hard to organize, and patients often have to wait for weekly or even monthly rehabilitation sessions in hospitals.

In the past few decades, the advances in technology have been increasingly exploited to implement modern approaches to rehabilitation. Computerized tools for cognitive training, providing exercises which focus on the basic cognitive domains (attention, memory, and executive functions) can be used in hospitals, as an alternative to the pen-and-paper approach described previously. These systems are not only more interactive, but also have the main advantage of being able to adapt the difficulty of the exercises according to the patient's personal history. Studies have found that the computerized approach is highly beneficial not only for patients but also for operators [21]: for example, computerized CR systems can log the patient performance and generate automatically summaries and comparisons for the operators. Additionally, there have also been tests for a home-based cognitive rehabilitation program based on traditional video games (specifically, Nintendo's Dr Kawashima's Brain Training for improving attention, processing speed, and working memory of patients with cognitive impairment due to multiple sclerosis [23]. Moreover, an italian research group developed Cognitive Training Kit (COGNI-TRAcK), an application for mobile devices, to self-administer an at-home, intensive, and personalized CR intervention based on working memory exercises. Upon testing its disposability-to-use (usability, motivation to use, compliance to treatment) on cognitive-impaired patients with multiple sclerosis, results showed an adherence of 84%; in fact, the participants felt highly motivated in performing the exercises, whilst the levels of perceived stress and boredom were low [25].

Below are listed some of the most used commercial cognitive rehabilitation software available in Italy [3]:

- **Brainer**: a web platform for cognitive rehabilitation developed by Brainer srl (`https://www.brainer.it`). Brainer has been used in multiple clinical studies, especially by patients suffering from schizophrenia. Brainer provides a higher number of exercises compared to the other systems, however they are not based on a realistic context. Furthermore, the difficulty must be manually selected by the operator and does not automatically adapt according to the patient's performance during the session.

- **Rehacom**: an advanced general-purpose software for cognitive rehabilitation, used specifically for multiple sclerosis, distributed by Hasomed (`https://www.rehacom.com`). Rehacom requires a 19" display, and can be enhanced by integrating a specific keyboard. Moreover, Rehacom provides exercises based on a realistic 3D context and also supports different languages. The software is available on a USB stick for at-home rehabilitation.

- **Erica**: a general-purpose software for cognitive rehabilitation developed by Giunti OS (`http://www.erica.giuntios.it`). Erica does not support at-home rehabilitation; all sessions must be performed at the hospital.

- **CogniPlus**: a scientific tool used for treatment of cognitive impairment developed by Schuhfried (`http://schuhfried.com`), available in 16 different languages. The software is available on a USB stick for at-home rehabilitation.

- **NeuroTracker**: NeuroTracker (`https://neurotracker.net/`) is a cognitive training program that is designed to improve mental performance. With NeuroTracker, each user is asked to complete a 3D visual exercise. In this exercise, users see 8 yellow balls moving around their screen. They are asked to track the highlighted balls. NeuroTracker's visual exercise uses 3D multiple object tracking technology which has been scientifically shown to enhance cognitive functions. To run NeuroTracker's program in anaglyph 3D, a user needs NeuroTracker's specialized 3D glasses, and a screen that is at least 15 inches wide is recommended.

- **Happy Neuron Pro**: HappyNeuron is a complete brain training method to stimulate the 5 main cognitive functions available to the general public and designed for people with no cognitive impairment; however, a Pro version is available (`https://www.happyneuronpro.com/`) which provides healthcare professionals with specifically adapted cognitive stimulation tools for use with their clients. The programs are implemented in a wide range of settings with users of various backgrounds and medical conditions (Cognitive Decline, Alzheimer's Disease, Behavioral Disorders, Stroke, Multiple Sclerosis, Schizophrenia...).

- **CogniFit**: CogniFit /`https://www.cognifit.com/`) is a healthcare company oriented to assess and improve cognitive health, providing scientifically validated exploration and brain function stimulation tools. One of the tools offered is CogniFit, a platform designed for healthcare providers so that they can assess, track, train their patients' cognitive skills. By simply creating an account, a clinician can send assessment and training recommendations to a large number of patients and access their results directly through the online platform. Specifically, a general cognitive assessment and several brain training programs are provided, so the clinician can decide which training fits the need of a particular patient. Clinicians can also create groups to manage several brain training patients at the same time.

1.2 The MS-Rehab platform

Multiple Sclerosis (MS) is a demyelinating disease, the most common autoimmune disorder affecting the central nervous system. It can damage several areas including movement, sensation, and visual, bodily, and other functions. Cognitive impairment has been recognized as a serious consequence of MS and cognitive problems are frequently observed (ranging from 43 to 70%) in patients suffering from MS [16]. MS-Rehab is an advanced cognitive rehabilitation system developed specifically for MS by Prof. Mauro Gaspari's research group of the Computer Science and Engineering Department of the University of Bologna, in collaboration with neurologists and psychologists of the Bellaria Hospital in Bologna, led by Doctor Sergio Stecchi. After having identified the limits of the pre-existing systems listed above, the aim of the project was to develop a computerized system able to integrate the various phases of cognitive rehabilitation processes, in order to improve cognitive rehabilitation procedures and optimizing them for routine clinical use in MS. Indeed, the existing computerized cognitive rehabilitation tools presented above, even if also used with MS patients, were born as general-purpose rehabilitation tools and thus not tailored for the specific disease. Moreover, the exercises are primarily used in the rehabilitation program phase, and the impact of the tool on the rest of the cognitive rehabilitation process is not explored in detail (such as automatic development of tailored, flexible and engaging activities using patients' cognitive profiles and collective rehabilitation cycles). The exercise developed for this thesis was designed for the MS-Rehab system. In the following section, after a brief overview of the cognitive rehabilitation process for MS patients, we will describe MS-Rehab's features and architecture.

	MS-rehab	Brainer	Reha com	Erica	Cogni plus	Neuro Tracker	Happy Neuron Pro	Cogni Fit
Target	MS Specific	General Purpose	General Purpose	General Purpose	General Purpose	General Purpose	General Purpose	General Purpose
Number of exercises	23	78	20+	35	15	1	38	33
Number of executive function exercises	4	16	4	13	2	1	12	16
Collective sessions	✓	✗	✗	✗	✗	✗	✗	✗
Home training	✓	✓	✓	✗	✓	✓	✓	✓
Semi-automatic conf.	✓	✓	✗	✗	✗	✓	✓	✗
Realistic content	✓	✗	✓	✗	✓	✗	✗	✗
Auto adaptive	✓	✗	✓	✓	✓	✓	✓	✓
Performance	✓	✓	✓	✗	✓	✓	✓	✓

Figure 1.4: Comparing MS-rehab with state-of-the-art commercial cognitive rehabilitation tools.

1.2.1 Cognitive Rehabilitation Process

Since one of the main goals was to build a system able to integrate the various phases of the MS cognitive rehabilitation process, a careful analysis was carried out of this process in practice in different Italian hospitals: the multiple sclerosis center of the Bellaria Hospital in Bologna, the multiple sclerosis centre of the Vaio Hospital, Fidenza (Parma), the Montichiari Hospital (Brescia), and the Neurological Clinic of Florence. The following phases of the cognitive rehabilitation process were identified:

- **Identification of the cognitive impairment**: an initial screening to identify patients with impairment caused by MS. The impairment may be noted due to the patient's own complaints, caregiver's reports, or the direct observation from the physician.

- **Multidimensional assessment of the impairment**: after excluding other possible causes, if the physician suspects that cognitive problems are an effect of MS, he may request a neuropsychological evaluation, in which a psychologist carries out a multidimensional assessment using various tools, such as clinical interviews, test batteries for the assessment of cognitive functions, and questionnaires to investigate the quality of life of MS patients.

- **Definition of the rehabilitation goals**: the team defines the rehabilitation goals based on the patient's specific needs in terms of home, education and work; the aim is to improve the overall quality of life of the patient, and in particular the personal, family-related and social aspects.

- **Definition of the rehabilitation program**: a rehabilitation program is established, following precise criteria: ecological tasks simulating real-life situations, rank-ordered interventions, intensity of treatment, and active involvement of patients throughout treatment. This program will then be carried out with individual or group sessions (where patients are grouped homogeneously according to their deficit) and home training.

- **Follow-up assessment of cognitive functions**: at the end of a rehabilitation cycle, a follow-up, including a clinical interview, test batteries, and a self-evaluation test, is arranged.

1.2.2 MS-Rehab features

The main purpose behind MS-Rehab was to build an advanced system specifically designed to support all the phases of the cognitive rehabilitation process described previously, as opposed to only the rehabilitation program phase, thus overcoming the limitations of the current tools. On account of this, the following requirements were kept in mind during the design of the system:

- **Full integration with the CR process**: MS-Rehab provides the ability to share data that is relevant for the rehabilitation program and making it available where necessary (for example, storing cognitive profiles of patients and using them for cognitive rehabilitation at home).

- **Centrality of the cognitive profiles of patients**: MS-Rehab offers flexible and semi-automatic customization tools for rehabilitation exercises based on patient profiles.

- **Full coverage of cognitive domains and rehabilitation activities**: MS-Rehab supports multiple forms of rehabilitation, such as individual exercises and group cycles in the hospital, and individual exercises at home, and allows to carry out these activities in a continuous and integrated form (for example, continuing at home the exercises started in the hospital).

- **Advanced monitoring of CR activities**: MS-Rehab includes functionalities for controlling and monitoring the rehabilitation and the patient's progress, such as statistics and other relevant data.
- **High usability for both clinical operators and patients**: MS-Rehab's interface has been not only carefully tested internally, but has also been subject to a formative usability study which adopted eGLU, a user-centered protocol including simplified usability tests, specific for finding critical issues in websites and web applications[14].

As a result, MS-Rehab is a comprehensive system which offers services for patients and operators to support the complete rehabilitation process.

Services dedicated to the patients The main service provided for the patient is, of course, the execution of the rehabilitation exercises. The user interface of MS-Rehab was deliberately designed to be minimal and intuitive, in order to stimulate the patient to carry out the exercises and encourage him to continue the training at home. Moreover, the patient can stop an exercise whenever he likes and restart later with an exercise at the same level of difficulty as the last one the patient has completed. In this regard, the variation of the exercises' difficulty is automatic: it is increased whenever the patient performance on an exercise exceeds 80% of the maximum score for two consecutive times. However, the automatic difficulty variation has lower priority than the operator's manual variations that can be set via the interface of the monitoring service.

Services dedicated to the operators MS-Rehab provides support for the operators in every phase of the rehabilitation process. For the multidimensional assessment phase (in which the psychologist establishes the patient's personal, clinical and neuropsychological profiles) the profiles can be inserted into the system and stored in a database (the CR data repository), in order to be accessed and updated at any time. For the rehabilitation program design phase, an intelligent configuration tool is provided for selecting the exercises of the rehabilitation sessions with the desired difficulty, according to the patient's cognitive profile and rehabilitation goals. The exercises are divided by category: attention, memory and executive functions. MS-Rehab also offers an interface to monitor the performance of the single patient or group of patients during the rehabilitation session (in hospital or at home), and increase or decrease the difficulty of the exercises accordingly.

1.2.3 MS-Rehab architecture

MS-Rehab is implemented as a web application, built on Apache Tomcat, which implements the Java Servlet and the Java Server Pages (JSP) specifications from Oracle Corporation. The user interface is built using Bootstrap, a popular HTML, CSS, and JavaScript framework for developing responsive, mobile-first web sites. This allows the system to be used via various devices, thus making it highly flexible: it can be accessed using either a PC or a tablet, and some services are even accessible by smartphone (such as the monitoring interface for individual and group sessions). The web server runs on a Ubuntu 16.04bvirtual machine and exploits Java 8 and Apache Tomcat 8.0.39. An SQLite database stores all the persistent data used by MS-rehab. Currently the MS-rehab server is hosted at the Computer Science and Engineering Department of the University of Bologna and it is accessible at the following address: `https://rehab.cs.unibo.it/MS-rehab-website/`. With the adoption of a traditional and robust web architecture, multiple rehabilitation centres can access the system and, if needed, the server can easily be migrated or replicated in any of them.

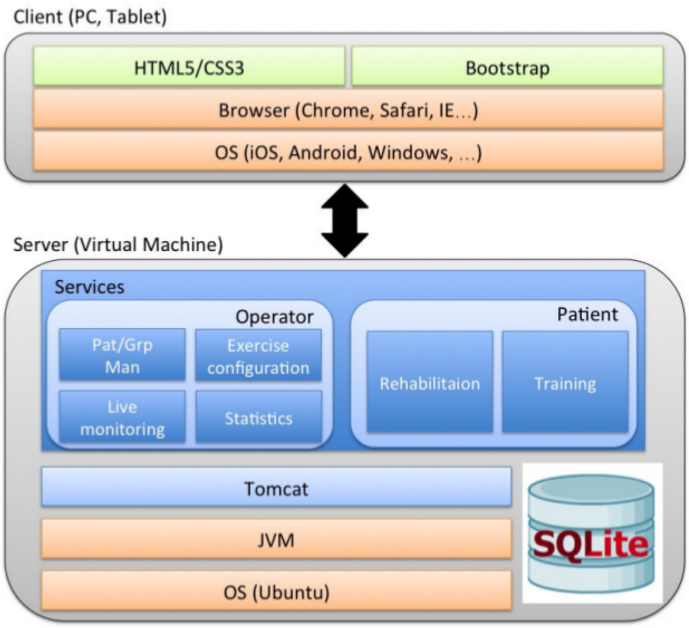

Figure 1.5: MS-Rehab architecture and technological stack [3]

Chapter 2

Automated Planning

Automated planning is a branch of artificial intelligence, related to decision theory, which involves devising a plan, described as a sequence of actions, in order to achieve a given goal. Typically, planning is used for realizing strategies to be executed by intelligent agents, autonomous robot or unmanned vehicle. In this chapter, we will give a brief introduction on classical planning and the PDDL language for formalizing planning problems.

2.1 Classical Planning

As stated previously, planning is one of the executive functions of the brain, and thus a fundamental property of intelligent behavior. It involves the explicit deliberation process that chooses and organizes actions by anticipating their outcomes, aiming at achieving some pre-stated objectives. Automated planning is the computational study of this deliberation process. In particular, plans are needed in many different areas of human endeavor and often it is useful, if not essential, to create these plans automatically. Automated planning technology now plays a significant role in a variety of applications, ranging from controlling space vehicles (such as the Mars Rover), managing fire extinctions or controlling underwater vehicles.

Specifically, given a description of the possible initial states of the world, a description of the desired goals, and a description of a set of possible actions, the planning problem is to synthesize a plan that, when applied to an initial state, is guaranteed to bring us to a goal state which contains all the desired goals. In the following paragraph, we will present a conceptual model [24] which allows us to describe in a more detailed manner the main elements of a planning problem and gain a basic understanding about the topic.

2.1.1 Conceptual Model

The following conceptual model of planning is composed by three primary parts:

- **State-transition system**: a formal representation of the real-world system we want to build plans for
- **Controller**: performs actions that change the state of the system
- **Planner**: produces the plans which drive the controller

Each of these parts will be discussed more in detail below.

State-Transition Systems

Definition. A state-transition system is a 4-tuple $\Sigma = (S, A, E, \gamma)$, where:

- $S = \{s_0, s_1, s_2, ...\}$ is a set of states
- $S = \{a_1, a_2, ...\}$ is a set of actions (state transitions whose occurrence is controlled by the plan executor)
- $E = \{e_1, e_2, ...\}$ is a set of events (state transitions whose occurrence is not controlled by the plan executor)
- $\gamma : S \times (A \cup E) \rightarrow 2^S$ is a state-transition function

The easiest way to represent a state-transition system is a directed graph where the nodes represent the states and the arcs represent the state transitions. Hence, if a state $s' \in \gamma(s, e)$, where $e \in A \cup E$ is an action or an event, then the graph will contain an arc from s to s' labeled with the action or event e.

If a is an action and $\gamma(s, a)$ is not empty, then we say that the action a is **applicable** to the state s: if the plan executor executes a in state s, the system will transition to some state s' in $\gamma(s, a)$.

The same can be said about an event e, with the slight difference that an event cannot be triggered by the plan executor and depends solely from the internal dynamics of the system. Consequently, if $\gamma(s, e)$ is not empty, we say that e *may possibly occur* when the system is in state s, and its occurrence will bring the system to some state s' in $\gamma(s, e)$.

A *planning problem* is formed by a state-transition system Σ as defined above, an objective (represented by a goal state s_g or a set of goal states S_g), and an initial situation (represented by an initial state s_0). The purpose of planning consists in finding a sequence of state transitions which starts in the initial state s_0 and ends in one of the goal states. Depending on the specific problem, the objective could also be to keep the system *away* from certain states, to optimize a utility function or perform a collection of tasks.

Planner

The second component of our conceptual model is the planner. A planner, given a planning problem P as input, outputs a *plan* (sequence of actions) which solves it. There are three main categories in which planners can be classified, according to the planning domains they work in:

Domain-specific planners As their name suggests, domain-specific planners are constructed and tuned specifically for a given planning domain and generally do not work in any other domains (or require major modifications in order to do so). Often, problem-specific techniques that are difficult to generalize to other planning domains are used. Most successful real-world planning systems work this way: for example. the Mars Rover planning software relied on a domain-specific planner.

Domain-configurable planners In this case, the planning engine is domain-independent but domain-specific knowledge is given as an input to the planner in order to constrain the search space.

Domain-independent planners These planning systems are built in order to be general enough so that they can work in any planning domain. All the planner needs as an input is the description of the planning problem to solve.

Since for this thesis a domain-independent planner was used, we will focus on the latter. As stated, in principle, this kind of planners works in any domain. However, in practice, restrictions are needed to obtain a trackableThe system. The trade-off for being domain-independent is efficiency, and in order to develop efficient planning algorithms, a set of simplifying assumptions are imposed on the domain. Specifically, this kind of configuration is used in *classical planning*. Classical planning domains have to satisfy the following assumptions:

1. **Finite** Σ: The system Σ has a finite set of states.

2. **Fully Observable** Σ: The system Σ is fully observable (i.e. has complete knowledge of the state of Σ

3. **Deterministic** Σ: The system Σ is deterministic (i.e. if an action is applicable to a state, its execution brings the system to a single other state, and similarly for the occurence of an event)

4. **Static** Σ: The system Σ is static (i.e.Σ has no internal dynamics and the set of events E is empty, thus the state changes only if an action is applied by the controller)

5. **Attainment goals**: The only kind of goal which is allowed is an attainment goal (i.e. a specific goal state or set of goal states which must be reached by applying a sequence of actions)

6. **Sequential Plan**: A solution plan is a linearly ordered finite sequence of actions

7. **Implicit Time**: Actions have no duration and are instantaneous state transitions

8. **Off-line Planning**: The planner receives no feedback about Σ's current state and is not concerned with any change that might occur in Σ while it is planning; it relies completely on a formal description of the system, the initial state and the required objective

Given these assumptions, we can define formally a classical planning problem as such: *Given* $\Sigma = (S, A, \gamma)$*, an initial state* s_0*, and a subset of goal states* S_g*, find a sequence of actions corresponding to a sequence of state transitions* $(s_0, s_1, ..., s_k)$ *such that* $s_1 \in \gamma(s_0, a_1), s_2 \in \gamma(s_1, a_2), ..., s_k \in \gamma(s_{k-1}, a_k)$*, and* $s_k \in S_g$*.*

It is immediately obvious that the limitations of classical planning restrict it to a very narrow class of planning domains. In fact, the imposed assumptions exclude most problems of practical interest, such as the previously cited example of the Mars Rover. However, as we will discuss in a later section, these assumptions fit well with the purpose of this thesis.

Controller

The third, and last, component of our conceptual model is the controller. The controller's input consists of plans and observations, a collection of sensor inputs which give information about the current state of the system. Supposing we have an observation function $\eta : S \to O$ which maps S into a discrete set of possible observations, the input to the controller will be the observation $o = \eta(s)$, where s is the current state. Observations can provide *complete* or *incomplete* information: in the first case, η is a one-to-one function, and thus from each observation o we can deduce precisely the current state. In the second case, η is not a one-to-one function, thus the only thing we can deduce from an observation o is that we are currently in one of the states of the set $\eta^{-1}(o) \subseteq S$. The controller's output consists in actions which will be performed in the state-transition system.

2.1.2 Representing the problem: modeling with PDDL

In 1998, in an attempt to standardize Artificial Intelligence planning languages, Drew McDermott and his colleagues proposed the Planning Domain Definition Language (PDDL) [27], inspired by STRIPS (STanford Research Institute Problem Solver) and Adl (Action Description Language). In this section, we will show how PDDL describes the key elements of a planning problem: states and actions.

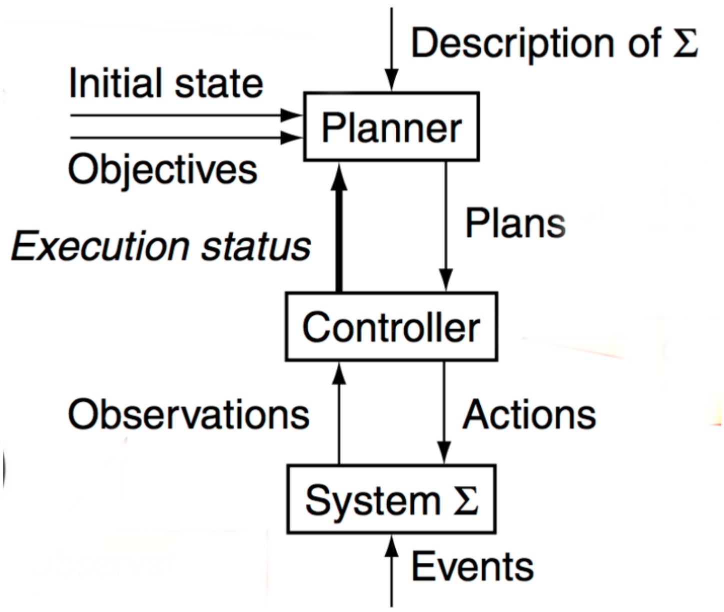

Figure 2.1: A planning system

State A state is represented as a conjunction of fluents that are ground, functionless atoms. Any fluent which is not mentioned is false, and their names are unique.

Action Actions are described by a set of action schemas. A schema specifies the name of an action, a list of all the variables used, a precondition (which defines the states in which the action can be executed) and an effect (the result of executing the action).

Specifically, planning problems are described in two files: a domain file, which describes the operators (or actions), and a problem file, which describes the initial state and the goal. In order to give a better idea of PDDL, we will use a classical planning domain example proposed at the first International Planning Competition [26] by S.Bartz and H.Kautz, `Logistics`. This domain describes a setting in which we want to move objects by planes and by trucks between different cities.

Domain description

The domain description consists of the following specifications:

Domain name A domain description always starts by the declaration if its name

Requirements Because PDDL is a very general language and most planners support only a subset, domains may specify the requirements, which define the expressive representation of the domain description: for example, whether action descriptions use typed terms, preconditions with equality terms, effects with conditional or universal quantifier terms...

Types PDDL allows the declaration of derived types (other than the primitive type `object`).

Predicates The predicates of a domain definition specify which predicates (denoting facts about the objects in the world) can be used in the preconditions and the effects of the operator and their parameters.

Operators Operators are the actions, i.e. how changes on the world state are performed. For each action, its name, parameters, preconditions and effects are specified.

```
(define (domain logistics)
    (:requirements
        :strips
        :equality
        :typing
        :conditional-effects
        :universal-effects)
(:types
    physobj - object
    obj vehicle - physobj
    truck airplane - vehicle
    location city - object
    airport - location)
(:predicates
    (at ?x - physobj ?l - location)
    (in ?x - obj ?t - vehicle)
    (in-city ?l - location ?c - city))
```

```
(:action load
    :parameters (?o - object ?v - vehicle ?l - location)
    :precondition (and (at ?o ?l) (at ?v ?l))
    :effect (and (in ?o ?v) (not (at ?o ?l))))
(:action unload
    :parameters (?o - object ?v - vehicle ?l - location)
    :precondition (and (in ?o ?v) (at ?v ?l))
    :effect (and (not (in ?o ?v))))
    ...
```

Listing 2.1: The Logistics domain file

In this example, a number of derived types are defined, such as *truck* and *airplane* which are derived from *physobj* which is in turn derived from the primitive type *object*. Parameters of predicates and actions are distinguished by their beginning with a question mark ("?"), and for each of them, its type is also stated. An example action in this domain is the *load* action, which has as parameters an object, a vehicle and a location. The preconditions are that the object and the vehicle must be at the location; and the effects are the the object will be in the vehicle and not at the location. Note that the vehicle can be either a truck or an airplane, as both those types are derived from the vehicle type.

Problem

The problem definition contains:

Domain The corresponding domain.

Objects The objects present in the problem instance.

Init The initial state description, i.e. a list of all the ground atoms that are true in the initial state. All other atoms are by definition false.

Goal The goal description, i.e. a formula of the same form as an action precondition. In difference to action preconditions, however, the goal descriptions should be ground.

```
(define (problem pb1)
    (:domain logistics)
    (:objects
        p1 p2 - obj
        london paris - city
        truck - truck
        plane - airplane
        north south east west - location
        lhr cdg - airport)
    (:init
        (in-city cdg paris)
        (in-city lhr london)
        (in-city north paris)
        (in-city south paris)
        (at plane lhr)
        (at truck cdg)
        (at p1 lhr)
        (at p2 lhr))
    (:goal
        (and
            (at p1 north)
            (at p2 south))))
```

Listing 2.2: A problem for the Logistics domain

In this example problem, two objects p1 and p2 are defined, along with two cities (London and Paris), a truck, an airplane, four locations corresponding to the the four cardinal directions, and two airports (LHR and CDG). In the initial states, the predicates state that CDG is in Paris, LHR in London, North is Paris, South is Paris, the plane is at LHR, the truck at CDG, and both of the objects are at LHR. The goal is to transport the first object p1 to North and the second to South.

Chapter 3

Weekend in Rome

The aim of this thesis is to design and implement a cognitive rehabilitation exercise for the rehabilitation of executive functions. Normally, this type of exercises require planning abilities, and are designed to have a high ecological value: the patient usually has to handle a situation he could encounter in his day-to-day life. Some examples of these exercises are already included in currently available cognitive rehabilitation software, but not without their limits, in particular in the number of available scenarios. In this case, the goal is to implement an exercise where there is a countless number of different scenarios, in the same vein as previously implemented planning exercises for MS-Rehab [4][5]. Thus, the patient is continually presented with new challenges, of increasing difficulty, forcing him to exercise his executive functions as effectively as possible.

In this chapter, we will describe the design and implementation of this exercise, its integration in the MS-Rehab system as well as a comparison with other existing exercises.

3.1 Design

The first step of the design phase of the exercise was laying the groundwork and devising the basic concepts of the exercise. This was followed by a precise analysis of the domain and actions in the exercise, and the formalization in PDDL. In this section, we will describe these two steps in a more detailed manner.

3.1.1 Groundwork

The initial objective was to formulate how the basic structure of the exercise. Weekend in Rome was inspired by one of the previously presented exercises for executive functions rehabilitation used at the Bellaria Hospital in Bologna *Map of Florence*: in this exercise, the patient is given a stylized map of Florence and a list of monuments or places to visit (for example, the Uffizi Gallery or the Piazza del Duomo), as well as some fixed-time appointments (for example, meet your friends for lunch at 1pm), and a starting time and location. The patient then has to plan the visit, choosing in which order to visit the monuments, taking into account the duration of the visits and the appointments. In this thesis, the setting was changed from Florence to Rome, and the time frame was increased from a single day to a weekend (two days). This allowed us to make the exercise even more realistic, thus increasing its ecological value, by adding the train and hotel reservations to the tasks the patient must accomplish in order to plan the vacation: the patient has to choose the train trip which allows him to arrive at the right time and the hotel located in the right place in order to to complete all the tasks.

Map and locations

At first, a few locations in Rome were selected among the most famous tourist attractions, chosen between two categories: locations which could be visited, and locations where an event could occur (such as a concert). The chosen locations were: the Trevi fountain, Saint Peter's Basilica, the Colosseum, the Pantheon, the Auditorium, the Ara Pacis, the Olympic Stadium and the Trastevere neighbourhood. The intent was to have some variety in the type of goals, which could range from the classic visiting a tourist attraction (such as the Trevi fountain), to attending a planned public event (such as a concert at the Auditorium or a football match at the Olympic Stadium) or a social occasion (such as having lunch with your friends in Trastevere). To increase the realistic aspect all the more, the patient is given two options for moving around the city: by foot, or by bus. In the latter case, bus timetables have to be taken into consideration as well.

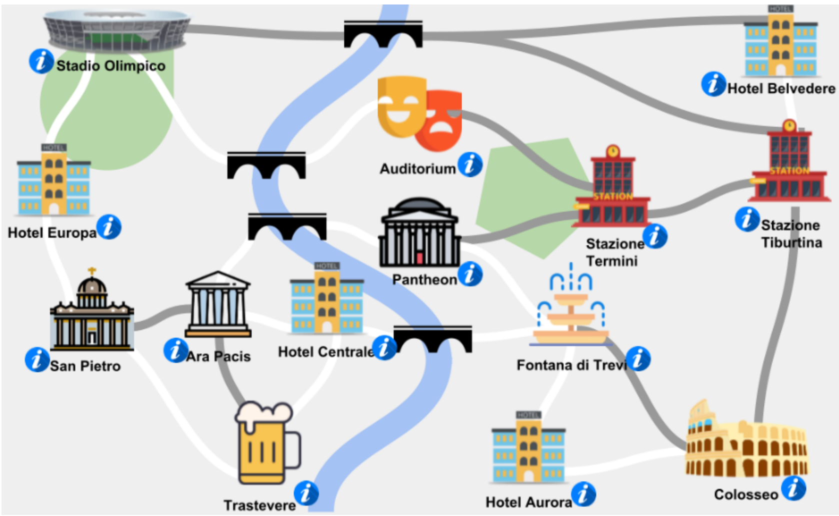

Figure 3.1: Weekend in Rome Map

At this point, a semi-realistic map with the chosen locations, the two train stations (Termini and Tiburtina), four hotels and possible bus routes was sketched in order to aid the modeling.

Goals

The next step was to choose the possible goals or tasks a patient had to accomplish. The immediate goals which could be defined were the train and hotel reservations. The latter imply that the patient has to sleep at the reserved hotel. To give a minimum of structure, it was decided to give a precise time at which the patient must go to sleep. Two additional tasks related to the hotel choice were defined to make the exercise more authentic: having breakfast at the hotel (taking into account breakfast hours), and exercising (doing gymnastics) before breakfast. In regards to the tasks pertaining to tourist activities, three main types were outlined:

1. **Simple sightseeing tasks**: tasks where it is just necessary to go to the specified location (for example, seeing the fountain of Trevi or the Colosseum).

2. **Activity tasks** : tasks where in addition to going to the specified location, an activity must be carried out (such as doing the guided tour of the Colosseum, or

visiting an exhibition at the Ara Pacis), taking into account the opening hours of the locations.

3. **Timed activity tasks**: tasks where an activity must be carried out at a specific time (such attending a football game at the Olympic Stadium, a concert at the Auditorium, or having lunch with friends in Trastevere).

At first, the idea was to generate the possible locations (and eventually, time of the event) randomly, in order to have countless goals to choose from. However, after a careful analysis, it was noted that some tasks only had sense if carried out in a few specific hours (for example, having lunch with friends in Trastevere at 5pm would be illogical). Thus, a list of specific tasks from which to choose from randomly was drawn up, with a total of 32 possible tasks (listed in table 3.1); the number was deemed sufficient to ensure that the exercises could be diversified enough.

1	Sleep at the hotel
2	Have breakfast at the hotel
3	See the Colosseum
4	See the Trevi fountain
5	See Saint Peter's Basilica
6	See the Ara Pacis
7	See the Pantheon
8	See Trastevere
9	Do the guided tour at Saint Peter's Basilica
10	Do the guided tour at the Colosseum
11	Get something to eat in Trastevere
12	Visit the exhibition at the Ara Pacis
13	Do the guided tour at the Pantheon
14	Have lunch with friends in Trastevere Saturday at 12pm
15	Have lunch with friends in Trastevere Saturday at 1pm
16	Have lunch with friends in Trastevere Sunday at 12pm
17	Have lunch with friends in Trastevere Sunday at 1pm
18	Have dinner with friends in Trastevere Saturday at 8pm
19	Have dinner with friends in Trastevere Saturday at 9pm
20	Meet up with friends in Trastevere Saturday at 10pm
21	Meet up with friends in Trastevere Saturday at 11pm
22	Meet up with friends in Trastever Sunday at 6pm
23	Meet up with friends in Trastever Sunday at 7pm
24	Watch the football game at the Olympic Stadium Saturday at 2pm
25	Watch the football game at the Olympic Stadium Saturday at 8pm
26	Watch the football game at the Olympic Stadium Sunday at 2pm
27	Watch the football game at the Olympic Stadium Sunday at 8pm
28	Go to the concert at the Auditorium Saturday at 8pm
29	Go to the concert at the Auditorium Saturday at 6pm
30	Go to the concert at the Auditorium Sunday at 7pm
31	Go to the concert at the Auditorium Sunday at 8pm
32	Do some gymnastics before breakfast

Table 3.1: Complete possible task list for the Weekend in Rome exercise

Use of automated planning

From the very beginning the integration of automated planning for the exercise's implementation was recognized as an essential requirement. One could argue that different approaches would have been possible: for example, given that the exercise basically consists in reaching a series of goals while satisfying certain constraints, one method which could come to mind is to solve the problem by modeling it as a constraint satisfaction problem (CSP). What makes automated planning a better solution is that while a CSP solver does find a solution in which all constraints are satisfied, a planner outputs the sequence of actions which can bring you to the solution, which is exactly what the patient is required to do in order to train his executive functions. Thus, with automated planning, we can compare the planner's solution to the patient's to have a performance metric. Moreover, the use of automated planning also allowed the implementation of a hint button (suggesting the next action to execute) and a verify button (giving the patient feedback on whether he is on the right track or not) as features of the exercise.

It is worth to note that automated planning has already been used previously in the context of software for cognitive impairment assistance: for example, in 2009, a canadian research group developed a planning aid to support clinical caregivers in the planning of daily living activities for patients with cognitive impairments. The tool also monitors the execution of a plan by the patient to generate reminders for forthcoming activities, alarms for anticipated failures, guidance and remedy in the ongoing activity [17].

3.1.2 Formalization in PDDL

Once the specification of the exercise was well defined, the next phase was modeling it in PDDL. At first, the domain was formalized, extrapolating the different possible actions and predicates. Subsequently, an example problem was defined, in order to build a first prototype to test the planner and specify a common basis on which to build the randomly generated problems. In the following paragraphs, this modeling phase will be described more in detail, with a particular attention to the modeling of time which had to be done explicitly.

Domain

The first step was to define the types of objects, that is the entities involved: in this case, two types were defined: `time` and `place`.

Modeling the domain also required a preliminary analysis to extrapolate the possible atomic actions in the different states. This resulted in the following list of actions:

- **Book train**: the patient will use this action to choose which trains to book. There are two possible situations: the booking of a single journey, or the booking of a round-trip. Every train trip has the duration of two hours (the same duration as the Bologna - Rome journey in real life).

- **Book hotel**: action to choose the hotel to book, among the 4 available.

- **Wait**: action to skip ahead in time of one hour.

- **Travel**: action to move around on the map. Two kind of travel actions were implemented; travel by foot and travel by bus. Both of the actions have a duration of one hour; the advantage of traveling by bus was built into the map layout (i.e. some bus routes act as a shortcut between two places where two or more travel by foot actions would be needed to get from one to the other).

- **Sleep**: action to go to sleep; two kind of sleep actions were implemented: a short sleep with the duration of four hours, and a long sleep with the duration of eight hours. These actions can only be executed at the booked hotel and at the determined sleep time, fixed by a specific predicate.

- **Have breakfast**: action (with the duration of one hour) to be done at the hotel after sleeping, during the breakfast hours of the hotel.

- **Do activity**: this action represents the different activities that can be done in the different locations (visit the exhibition, do the guided tour, watch the football match...). For simplicity, it was determined that all the activities have the duration of two hours.

- **Exercise**: this action has the duration of one hour, and must be done before breakfast.

During the definition of the actions, the needed predicates to express accurately the preconditions and effects emerged as well. In particular, predicates for the following notions were defined:

- **Train predicates**: These predicates are used to define the possible outward, return or round-trip journeys that are available for booking.

- **Map predicates**: These predicates are used to model the underlying map of Rome and its mechanics, such as `foot-path` and `bus-path` to specify which locations are accessible with which kind of transport, `opening-hours` to define the opening hours of the different locations, `can-sleep` and `train-station` to denote whether a location is a hotel or a train station, `breakfast-hours` to define the hours during which breakfast is available at the hotel, and `activity-available` to mark the locations where an activity is available.

- **Predicates to keep track of goals**: These predicates are used to keep track of the actions that have already been done, in order to identify when all the goals have been satisfied. Some examples are `slept`, `visited`, `done-activity`, texttt done-activity-timed and `done-exercise`.

- **Predicates to model time**: These predicates will be discussed in detail in the next paragraph.

Modeling of time

As mentioned previously, time has to be modeled explicitly. This choice was also due to the solver (discussed specifically in the following paragraphs) not supporting some features of PDDL 2.1 that would have helped in modeling time, namely numeric fluents (for modeling non-binary resources) or durative actions (which could have variable, non-discrete length, conditions and effects).

The basic idea our representation of time is that each atomic time unit is represented by a `time` object. In the case of Weekend in Rome, 48 `time` objects were defined, corresponding to the 48 hours of the two-day weekend. A chronological order is established using predicates, and the passage of time is modeled by marking these objects as in the past, present or future. To model this representation, the following predicates were defined:

- `(consecutive ?hour1 ?hour2 - time)`: to establish the order in which the different hours are in sequence
- `(future ?time - time)`: to keep track of past and future hours
- `(at ?p - place ?time - time)`: to keep track of present time and location

Additionally, as the reservations compulsorily had to be the first two actions, it was also necessary to define predicates to force the order of the actions by using them in the preconditions of the other actions: `done-train-booking` and `done-hotel-booking`.

Two examples of actions formalized using PDDL and the solver's syntax are described below.

```
(:action travel-by-bus
   :parameters (?src ?dst - place ?departure-time ?arrival-time - time)
   :precondition (and
                   (done-hotel-booking)
                   (at ?src ?departure-time)
                   (consecutive ?departure-time ?arrival-time)
                   (bus-path ?src ?dst)
                   (future ?arrival-time)
                   (bus-scheduled ?src ?dst ?departure-time)
                 )
   :effect (and
               (not (future ?departure-time))
               (not (future ?arrival-time))
               (not (at ?src ?departure-time))
               (at ?dst ?arrival-time)
               (visited ?dst)
           )
```

```
)
```

Listing 3.1: Travel by bus action formalized in PDDL

The action in 3.1 specifies that you can travel by bus from a location `?src` at the hour `?departure-time` and to a location `?dst` arriving at the hour `?arrival-time`. The preconditions dictate that:

- the patient has to have finished the reservations
- the patient has to be in the start location at the departure time
- the arrival time must be hour immediately following the departure time
- there must be a bus path between the two locations
- the arrival time must be in the future
- there must be a bus scheduled on that route at the departure time

The effect equation says that, once the action is applied, you are at the destination at the arrival time, are no longer at the source location at the departure time, the departure time and the arrival time are in the past (i.e. no longer in the future), and the destination is marked as visited.

```
(:action do-activity
:parameters ( ?p - place
              ?hour1 ?hour2 ?hour3
              ?opening-time ?closing-time - time)
:precondition (and
                (done-hotel-booking)
                (at ?p ?hour1)
                (activity-available ?p ?hour1)
                (consecutive ?hour1 ?hour2)
                (consecutive ?hour2 ?hour3)
                (future ?hour2)
                (future ?hour3)
                (opening-hours ?p ?opening-time ?closing-time)
```

```
                    (not (future ?opening-time))
                    (future ?closing-time)
                    (not (= ?hour2 ?closing-time))
                    (not (done-activity ?p))
                  )
  :effect (and
                (at ?p ?hour3)
                (not (at ?p ?hour1))
                (done-activity-timed ?p ?hour1)
                (done-activity ?p)
                (not (future ?hour1))
                (not (future ?hour2))
                (not (future ?hour3))
          )
  )
```

Listing 3.2: Do activity action formalized in PDDL

The action in 3.2 specifies that you can do an activity in the location `?p`, which is open from `?opening-time` to `?closing-time`, starting at the time `?hour1` up to the time `hour3` (indeed, as stated previously, actions having the standard duration of two hours). The preconditions dictate that:

- the patient has to have finished the reservations
- the patient has to be in the start location at the starting time
- there must be an activity available at the location
- the activity must take place during consecutive hours
- the middle hour and the ending time of the activity must be in the future
- the activity must take place during the opening hours of the location
- the activity must not have been done before

The effect equation says that, once the action is applied, you are at the location at the ending time, are no longer at the location at the starting time, the starting time, middle hour, and ending time are in the past, and the destination is marked as done.

Problem

To focus on building an initial prototype, a fixed, sample problem was defined as a basis for implementing the back-end and front-end of the exercise. This sample problem also helped to identify a common structure for all the problem instances, on which to build on with the exercise generator. Specifically, the common basis consisted in:

- **Objects** - the objects involved are the same for each problem: the locations and the 48 hours
- **Initial state** - most of the initial state description is common to all instances, namely the predicates to model the map and to model time.

```
(define (problem default-problem)
  (:domain weekend-in-rome)
  (:objects
        hour1 hour2 hour3 [...] hour47 hour48 - time
        termini pantheon tiburtina [...] - place
  )
  (:init
    ; Time
    (consecutive hour1 hour2)
    (consecutive hour2 hour3)
    [...]
    (consecutive hour47 hour48)
    (future hour1)
    (future hour2)
    [...]
    (future hour48)
```

```
    ;Map
    (foot-path stadio-olimpico europa)
    (foot-path europa stadio-olimpico)
    (bus-path termini tiburtina)
    (bus-path tiburtina termini)
    [...]
    (bus-scheduled colosseo tiburtina hour5)
    (bus-scheduled colosseo tiburtina hour7)
    [...]
    (can-sleep aurora)
    (can-sleep belvedere)
    (can-sleep centrale)
    (can-sleep europa)
    (activity-available san-pietro hour10)
    (activity-available san-pietro hour11)
    [...]
    (train-station termini)
    (train-station tiburtina)
    (opening-hours colosseo hour9 hour18)
    (opening-hours ara-pacis hour14 hour18)
    [...]
    (breakfast hour32 hour35)
    (at-bologna)
)
```

Listing 3.3: Common structure for all problem instances

3.2 Implementation

Once the model was formalized in PDDL, the next step was to build a functioning prototype of the exercise based on a the default sample problem. In the following paragraphs we will discuss the realization of the front-end and back-end of the application.

The front-end consists in what the user sees and interacts with - in this case, the map and control panel. The back-end usually handles business logic and data storage. In this case, the back-end is responsible for generating the exercises, and calling the planner when necessary.

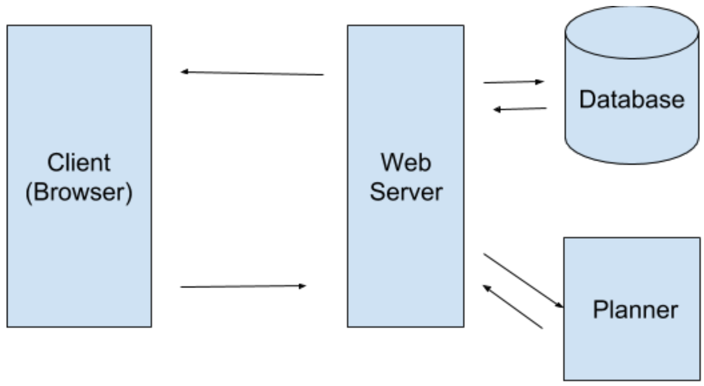

Figure 3.2: Front-end and Back-end of the application

3.2.1 Front-end and user interface

The front-end was built using HTML5 (to be embedded in the JSP) and Javascript. To handle all the different locations an array of objects was created where each element represented a location and all the information relative to it. For example, in 3.4, the object which represents all information relative to the Colosseum is shown.

```
var colosseo = {
    id:0,
    name: "Colosseo",
    pddlName: "colosseo",
    texture: PIXI.Texture.fromImage('resources/colosseo.png'),
    coordinates: {
        x: 0.89,
        y: 0.85
    },
    adjacentLocationsBus: [13,1],
    adjacentLocationsWalk: [7],
```

```
    busSchedules: {
        13: [5,7,9,11,13,15,17,19,21,23,25,31,34,37,40,43,46],
        1: [8,11,14,17,20,24,32,36,40,44]
    },
    openingHours: {
        saturday:[9,18],
        sunday:[38,42]
    },
    activity: ["Visita"],
    currentlyAccessible: false,
    doneActivity: false
};
```

Listing 3.4: The object corresponding to the Colosseum

Each location object has the following properties:

- **id** : a number ID which represents the location
- **name** : the location name
- **pddlName**: the name of the object which represents this location in the PDDL problem (for builing the PDDL strings of the actions to send to the server)
- **texture**: the image file of the icon of the location
- **coordinates**: the coordinates of the location on the map
- **adjacentLocationsBus**: the IDs of the locations which are accessible by bus from this location
- **adjacentLocationsWalk**: the IDs of the locations which are accessible by foot from this location
- **busSchedule**: for each location which is accessible by bus, an array is stored with the hours in which there is a bus available for that location
- **openingHours**: the opening and closing time of that location

- **activity**: the activities available in that location (if any)
- **currentlyAccessible**: whether the location is accessible in the current position of the user or not (to handle movement around the map)
- **doneActivity**: whether the activity available in this location has already been done

Two features which require to interact with the planner, named "Verify" and "Hint", were implemented : the "Verify" feature will give the patient an input on whether the problem is still solvable or not. In the latter case, it will inform the patient of which goal is unsolvable and give him the options to either continue in order to satisfy the other goals, or end the exercise. The "Hint" button instead will give the patient a hint on which action to perform next. For both of these features, it is necessary to keep track of all the user actions in order to update the problem state. This is done by initializing a `userActions` array which would be updated with a properly formatted PDDL string after each executed action. This array is then passed with a HTTP POST request to the server whenever the Verify or Hint features are used.

Figure 3.3: Verify button result

For each possible action, a corresponding method was implemented which would update the current time and the `userActions` array. In 3.5, the method which is called when the user clicks the "Wait" button is depicted. To make the exercise a bit more realistic, when the patient decides to perform an action by clicking on its corresponding button, a loading dialog is shown for a few seconds, to simulate the fact that time is passing. Then, the time is updated (in the case of the "Wait" action, it is incremented by one, since the "Wait" action has a duration of one hour), the control panel buttons are updated and the PDDL string corresponding to the action with the correct parameters is built and added to the `userActions` array.

```
$("#waitButton").click(function(){
    // Waiting dialog
    var dialog = bootbox.dialog({
        message: 'Aspettando 1 ora...',
        closeButton: false
    });
    // Save start time for building PDDL string and update time
    var startTime = currentTime;
    setTimeout(function(){
        // Update date and time
        currentTime++;
        updateTime();
        dialog.modal('hide');
        var sourceId = findClickedLocation(user.x, user.y);
        var currentLocation = getLocationFromId(sourceId);
        updateButtons(currentLocation);
    }, 2000);
    // Build PDDL String and add user action to user actions list
    var endTime = startTime + 1;
    var actionString = "(wait " + currentLocation.pddlName + " hour"
                        + startTime + " hour" + endTime + ")";
    userActions.push(actionString);
});
```

Listing 3.5: The wait action method

User Interface

The user interface is separated in two areas: the control panel, where the current time and all the buttons to access information or to execute the possible actions are displayed, and the map area. Initially, the layout was designed as two columns - one with the map, and one with the control panel: however, in a subsequent phase, it was deemed better for the overall usability to make use of as much space as possible for the map, especially considering the app is mainly designed to be accessed by tablet. Thus, the layout was changed so as to integrate the control panel components in the top bar, leaving all the underlying space for the map, as in figure 3.4.

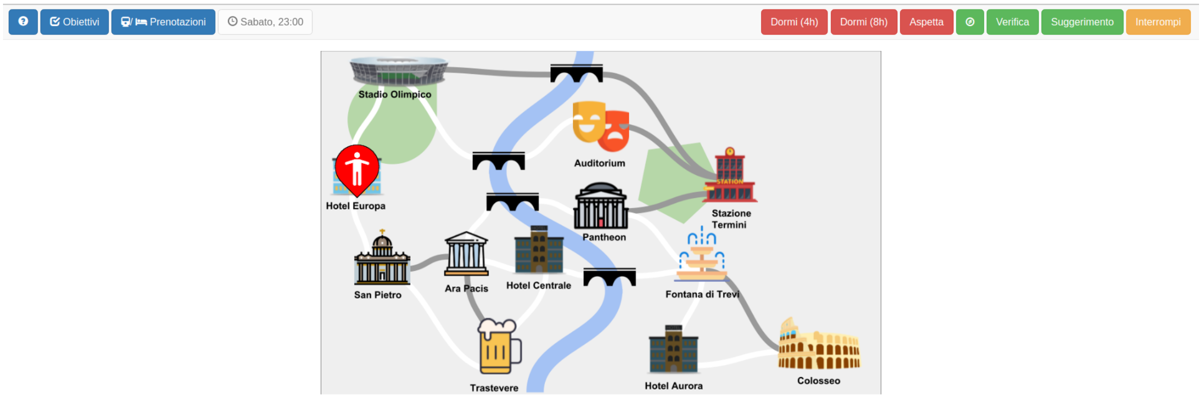

Figure 3.4: Control Panel with Hotel actions

Control Panel For the graphic design of the control panel the Twitter Bootstrap 3.0 framework was used to keep it consistent with the the rest of the interface of the MS-Rehab system. The control panel displays the current date and time in the exercise, and various buttons according to the current location and state of the exercise. Specifically, there are three categories for the buttons:

- **Information buttons** - these buttons are always present during the exercise. There are three of them: the instructions button, which displays the modal with the instructions of the exercise and map legend; the goals button, which display the list of goals to be achieved and whether they are satisfied, insatisfiable or to be satisfied; and the reservations button, which displays the details of the reserved trains and hotel so the patient does not have to remember them and can concentrate on the planning aspect of the exercise.

- **Action buttons** - these buttons change during the exercise. At first, the only action is available is ≪Book Trains≫, followed by ≪Book Hotel≫. Once the reservations have been made, the button ≪Depart≫ appears: once clicked, the current location icon appears on the map and the patient can start moving around. In this phase of the exercise, the action buttons depend on the current location and display the possible actions in that moment at that locations (for example, visiting the exhibition when at the Ara Pacis or sleeping when at the hotel).

- **Help buttons** - these buttons are always present in the second phase of the exercise (after the reservations have been made). There are three help buttons available for the patient: the first is a "compass", which, once clicked, will display a red circle around the locations the patient con access from his current location, along with an icon representing the corresponding type of transport (by foot or by bus). The other two buttons are the "Verify" and "Hint" buttons discussed previously, which involve an HTTP request to the server to interact with the planner.

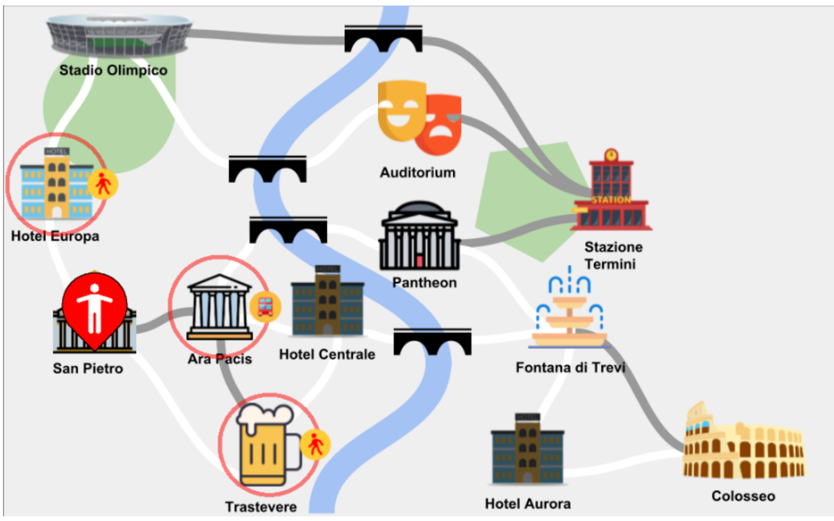

Figure 3.5: Weekend in Rome Map with compass hints

In order to facilitate the patient and make the interface more intuitive, a color code was used for differentiating each category of buttons: blue for information buttons, red for action buttons, and green for help buttons.

Map To implement the map, depicted in figure 3.1, the Pixi.js library was used. Pixi.js is an open source, cross browser JavaScript 2D WebGL graphics library with canvas fallback. Since the map is essentially static, a background with the roads was drawn and loaded as a 1000x628 PixiJS sprite. The different locations are designated by colourful icons, added programmatically as Pixi.js sprites. The library also allowed to handle click and touch interaction easily using the `interactive` and `buttonMode` properties of sprite objects.

3.2.2 Back-end

Bearing in mind it would be integrated into the MS-Rehab system, the back-end of the exercise was built in Java 1.8 and using the PDDL4J library, an open source library under LGPL license whose purpose of PDDL4J is to facilitate the development of JAVA tools for Automated Planning based on PDDL language [18].

The structure of the server side of the Weekend In Rome exercise can be described by the following class diagram:

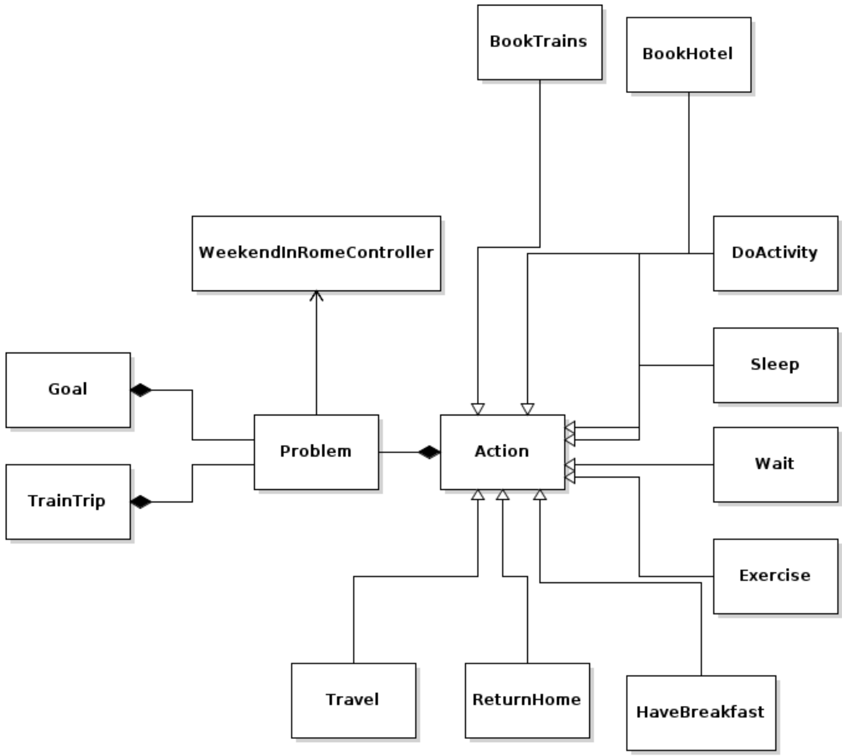

Figure 3.6: Class Diagram

The `WeekendInRomeController` class handles the planning aspect. It has three main responsibilities:

- Generating the problem
- Checking if the problem is solvable
- Getting the first action of the solution plan to a problem

These three responsibilities correspond to the three instances in which the planner is used: when the Verify feature is used, when the Hint feature is used, and during problem generation.

An instance of the `Problem` class is created during problem generation, just before the patient starts the exercise. This `Problem` object represents the problem that must be solved in that specific instance of the exercise and contains all the information which characterizes it. In particular, goals (i.e. the tasks that the patient must accomplish) and available train trips are represented by the `Goal` class and the `TrainTrip` class respectively. Additionally, a superclass `Action` was created with a subclass for each possible action. These classes are instantiated when the current problem state of the planner must be updated with the patient's actions.

Exercise Generator

The `generateProblem` method of the WeekendInRomeController, given the desired level of difficulty, generates a random problem. The possible goals are divided into arrays according to their type (described in 3.1.1), and are chosen randomly. The number and type of goals depends on the difficulty level. Once the goals are generated, a time to go to bed is chosen accordingly and then the possible train trips are generated. At this point, a `Problem` object is generated and is given in input to the `isSolvable` method. If the problem is not solvable, the process is repeated until a solvable problem is generated.

Verify and Hint features

The complexity of implementing the Verify and Hint features lies in the fact that the patient may execute a number of actions before calling either of those methods. This means that the problem state of the planner must be updated with relevant data whenever one of these features is used. In order to do so, an abstract class `Action` was created with the `generateEffects` method. Then, for each type of action, a corresponding class was created which extended the `Action` class, implementing this method depending on the effects of the specific action. When the `WeekendInRomeController` receives from the front-end the list of user actions, it parses each string and creates the corresponding action instance setting the specific parameters of that action. Then, the `applyAction` method of the `Problem` class is called on each `Action` instance. This method calls `generateEffects` method which will output the list of effects in PDDL syntax with the right parameters. Finally, this list is iterated through and each effect is either added to the beliefs list or removed (if it is a negated effect). Once the problem is updated, the `WeekendInRomeController` checks if its is solvable and returns a boolean response in case of the Verify feature, or the first action of the solution plan in case of the Hint feature.

Varying difficulty

Once the prototype was completed, the next issue to face was to define how to vary the difficulty level of the exercise. In MS-Rehab, there are three categories of difficulty: easy, medium, and hard. Each category has a number of sub-levels which, in their turn, increase slightly in difficulty. After a careful analysis, with the help of trained psychologists, the following difficulty levels were defined:

- Easy:
  - Simpler map (only one train station and three hotels), pictured in 3.7
  - Same opening hours for every location
  - Buses are available at every hour

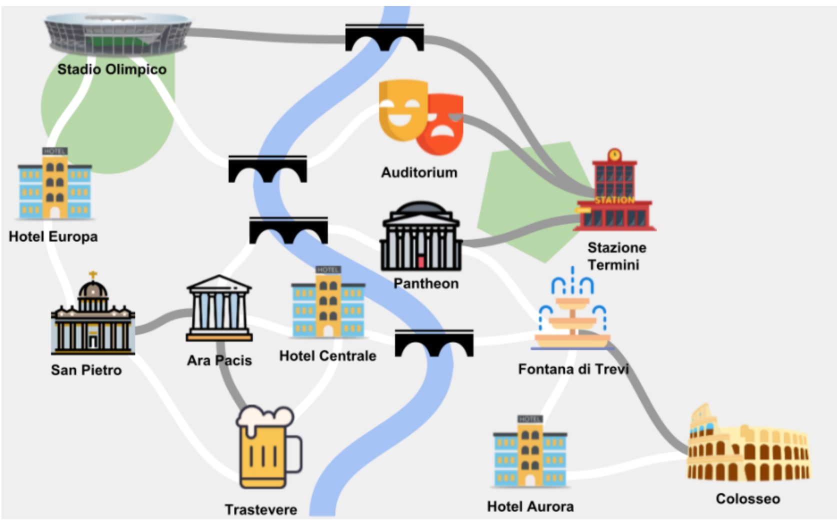

Figure 3.7: Weekend in Rome Map: Easy Difficulty

- Medium:
  - One train station is added (but the hotels remain three), pictured in 3.8
  - Different opening hours for every location
  - Buses have schedules

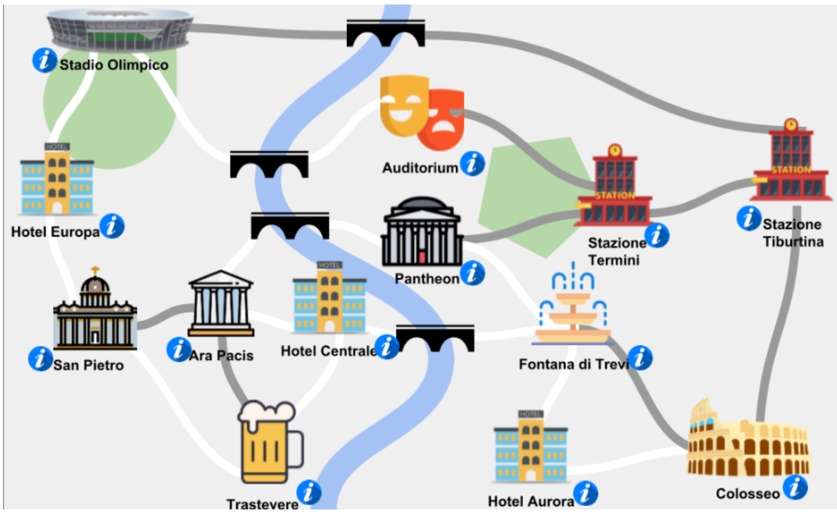

Figure 3.8: Weekend in Rome Map: Medium Difficulty

- Hard:
  - Original map (two train stations and four hotels), pictured in 3.1
  - Different opening hours for every location
  - Buses have schedules

Additionally, for each sub-level of difficulty, goals were generated according to the following schema:

Difficulty	Level	Sightseeing Goals	Activity Goals	Timed Activity Goals
EASY	1	2	0	0
	2	2	1	0
	3	2	1	1
MEDIUM	4	1	1	0
	5	2	1	0
	6	2	1	1
HARD	7	0	1	1
	8	1	1	1
	9	0	2	2

Table 3.2: Goal schema for difficulty levels

The first sub-level of each category (easy, medium or hard) always has two goals to accomplish; the second, three goals and the third four goals. What changes between the categories is the type of goals: indeed, sightseeing goals are the easiest to complete as there are no time constraints. Activity goals are slightly harder, as the two-hour duration must be taken into account as well as the location's opening hours. Finally, timed activity goals are the hardest as they can only be completed at a specific time. Moreover, for medium and difficult levels, the "have breakfast at the hotel" goal was added to all problem instances, as was the "exercise before breakfast" goal for all difficult levels. This means that the total number of goals rises from two at level 1 to six at level 9. This choice was also influenced by the planner, as the more goals are added (especially in the case of timed activity goals), the more time it would take to generate a solvable problem, thus resulting in a longer loading time.

3.3 Comparison with other exercises

In the following section we will compare the Weekend In Rome exercise with other similar exercises for the rehabilitation of executive functions, namely the two currently available planning exercises in the MS-Rehab system (*Una mattina di impegni* and *Visita allo ZooSafari*), and the *Plan a HoliDay* exercise of the RehaCom cognitive training software.

Una mattina di impegni (MS-Rehab) This exercise is based on the Plan-A-Day exercise proposed by Funke & Krüger in 1995. In this exercise, the patient has to schedule a list of tasks to complete during the day (for example, picking up his daughter from the swimming pool or buying groceries) while considering various constraints about when, where, and for what duration the activities have to be carried out. Conceptually, the exercise is similar to Weekend In Rome: the patient has to move around on a map and execute given tasks. However, in Weekend in Rome, there is the additional difficulty of having to plan the train and hotel reservations beforehand. In regards to the implementation, one main difference is that in *Una mattina di impegni*, the planner is called after every action of the patient: if the new problem has a solution, the patient may continue, otherwise the exercise ends. This approach was not adopted for Weekend In Rome as there was a risk of having long loading times while the planner searched for a solution, making the exercise slow and bothersome. Another difference is that in the case of Una mattina di impegni, a new map is generated for each exercise; in Weekend in Rome, there are 3 static maps (corresponding to the 3 difficulty levels). The logic for moving around on the map was also changed from drag & drop to clicking, as the first proved to be not easy to use for patients on tablet.

ZooSafari (MS-Rehab) ZooSafari was the second planning exercise implemented for MS-Rehab, inspired by the Zoo Map test. In this exercise, the patient has to plan his route through a map of the zoo, visiting a selection of locations while actively disregarding others and obeying certain rules: for example, the order of the locations to visit, or crossing certain paths only once. In this case the approach used in Una Mattina di Impegni (calling the planner after each action of the patient) was also avoided. However, one big difference is that Weekend in Rome offers two additional features: the Verify feature and the Hint feature.

Plan a HoliDay (RehaCom) A similar planning exercise is available in the RehaCom therapy system in the Plan a Vacation module, under the name of Plan a HoliDay [20]. Despite the name, the exercise resembles more *Una mattina di impegni* rather than

Weekend in Rome: the patient's ability to organize and plan a day is trained, in a realistic manner (particular tasks have to be dealt with at specific places and must be completed within a given point in time). Thus, the exercise does not included train or hotel reservations, and plans are made only for one day - unlike Weekend in Rome. However, Plan a HoliDay also provides an almost endless number of different tasks since new combinations of tasks can be generated randomly.

Figure 3.9: Plan a HoliDay exercise in the RehaCom system

Conclusions

The rehabilitation of esecutive functions is essential for improving the quality of life of patients affected by multiple sclerosis, as their impairment compromises even the simplest day-to-day activities such as planning a meal or dressing oneself. If it is possible to improve executive functions through rehabilitation, then more people with MS might become more independent with activities of daily living, and might respond better to their rehabilitation.

MS-Rehab offered three exercises for the rehabilitation of executive functions - one on inhibition control, and two other which involved planning aspects (*Una mattina di impegni* and *Visita allo ZooSafari*). The objective of this work was the design and implementation of an additional exercise for the rehabilitation of executive functions and its subsequent integration in the the MS-Rehab system. In this exercise, the patient has to plan a two-day vacation in Rome by reserving the trains and hotel, and planning the itinerary in the city while taking into consideration a number of constraints, mirroring the kind of limitations one would have to plan for in real life (such as opening hours and bus schedules). This leads to the first key aspect of the exercise: its high ecological value. Indeed, while there are quite a few computerized cognitive rehabilitation exercises which involve planning a day into the city, none of them take place in a two-day span nor include the train and hotel reservations. A second important aspect is the generation of constantly new problems by combining randomly the different tasks to be carried out. Careful attention was also given to the user interface, in effort to make it not only easy to navigate and intuitive for the patient, but also pleasurable and fun in order to encourage the patient to carry out the exercise even at home. Last but not least, a novel feature which is not present in the other two planning exercises of the MS-

Rehab system was implemented: the Hint feature, in which the patient is suggested the next action to execute in order to carry out all the given tasks. The varying difficulty levels were carefully defined in collaboration with a psychologist specialized in cognitive rehabilitation, which also deemed the exercise suitable for its planned purposes, that is the rehabilitation of executive functions.

Future work

A possible extension of the exercise would be to add a budget constraint, implemented through numeric fluents in PDDL (which are not currently supported by the PDDL4J planner). The exercise could also be enriched with ulterior activities to be carried out as tasks, such as a boat ride on the Tiber, or a stroll in the Villa Borghese park. Maps representing other cities, such as Florence or New York, could be designed for variety.

Another possible, but more challenging extension, would be to implement a 3D version of the exercise, in which the patient has to navigate through the streets of the city in first-person view, eventually exploiting Virtual Reality technologies. Indeed, VR offers the potential to develop human testing and training environments that allow for the precise control of complex stimulus presentations in which human cognitive and functional performance can be accurately assessed and rehabilitated. Although further study is needed, some studies have shown that some applications are effective in treating cognitive deficits in people with neurological diagnoses [28].

List of Figures

Listings

Bibliography

[1] Jongen PJ, Ter Horst AT, Brands AM. *Cognitive impairment in multiple sclerosis.* Minerva Med. 2012;103:73–96.

[2] Schulz D, Kopp B, Kunkel A, Faiss JH. *Cognition in the early stage of multiple sclerosis.* J Neurol. 2006;253:1002–10.

[3] 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.

[4] Bartolomeo Lombardi, *La pianificazione per la riabilitazione delle funzioni esecutive.* Tesi LM, Università di Bologna, Scuola di Scienze, Corso di laurea in Informatica. 2018.

[5] D. Baschieri, *Riabilitazione cognitiva delle funzioni esecutive nella Sclerosi Multipla: un approccio basato sul planning.* Tesi LM, Università di Bologna, Scuola di Scienze, Corso di laurea in Informatica. 2017.

[6] John L. R. Rubenstein, Pasko Rakic, *Neural Circuit Development and Function in the Brain: Comprehensive Developmental Neuroscience*, Volume 3 of Comprehensive developmental neuroscience, Neural Circuit Development and Function in the Healthy and Diseased Brain, Academic Press, 2013.

[7] Baddeley, A., & Wilson, B., *Frontal amnesia and the dysexecutive syndrome.* Brain and Cognition, 7, 212–230. 1988.

[8] Institute of Medicine, *Cognitive Rehabilitation Therapy for Traumatic Brain Injury: Evaluating the Evidence*, The National Academies Press, 2011.

[9] Diamond A., *Executive Functions.* Annual review of psychology.64:135-168. 2013.

[10] Adel C. Najdowski, Evelyn Kung, *Evidence-Based Treatment for Children with Autism*, Academic Press, 2014.

[11] 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.

[12] McKay Moore Sohlberg and Catherine A. Mateer, *Cognitive Rehabilitation: An Integrative Neuropsychological Approach*, 2001.

[13] Halligan, P.W., & Wade, D.T. (Eds.) *Effectiveness of Rehabilitation for Cognitive Deficits.* Oxford University Press, UK, 2005.

[14] F. Zini, E.M. Bressan, M. Gaspari, *A formative user–based usability study for an advanced cognitive rehabilitation system.* In Proceedings of CHItaly '17, Cagliari, Italy, Septe

[15] Oosterman, Joukje & Wijers, Marijn & Kessels, Roy, *Planning or Something Else? Examining Neuropsychological Predictors of Zoo Map Performance*, Applied neuropsychology, Adult, 20, 103-9, 2013.

[16] R. H. Benedict and R. Zivadinov, *Risk factors for and management of cognitive dysfunction in multiple sclerosis*, Nature Reviews Neurology, vol. 7, no. 6, pp. 332–342, 2011.

[17] La Placa, M., Pigot, H., & Kabanza, F. *Assistive planning for people with cognitive impairments.* In Proc. of Workshop on Intelligent Systems for Assisted Cognition hosted by Int'l Joint Conference on Artificial Intelligence (IJCAI). 2009.

[18] Pellier, D., Fiorino, H. *PDDL4J: a planning domain description library for Java.* Journal of Experimental & Theoretical Artificial Intelligence, 30(1), 143-176. 2018.

[19] Kroll, J., Karolis, V., Brittain, P. J., Tseng, C. J., Froudist-Walsh, S., Murray, R. M., & Nosarti, C. *Real-life impact of executive function impairments in adults who were born very preterm.* Journal of the International Neuropsychological Society, 23(5), 381-389, 2017.

[20] RehaCom User Manual - Plan a Vacation. Hasomed. 2015.

[21] A. Compston and A. Coles. *Multiple sclerosis.* Lancet, 372(9648):1502-1517,Oct 2008.

[22] Adrian M. Owen, *Cognitive planning in humans: Neuropsychological, neuroanatomical and neuropharmacological perspectives*, Progress in Neurobiology, Volume 53, Issue 4, Pages 431-450, 1997.

[23] Prosperini L, De Giglio L, Pozzilli C, Pantano P, Bianchi V, Borriello G. et al. *A Low-Cost Cognitive Rehabilitation With a Commercial Video Game Improves Sustained Attention and Executive Functions in Multiple Sclerosis: A Pilot Study.* Neurorehabilitation and Neural Repair. 2014.

[24] Dana DS. Nau, *"Current trends in Automated Planning"*, AI Magazine, 28(4), 43, 2007.

[25] A. Tacchino, L. Pedulla, L. Bonzano, C. Vassallo, M. A. Battaglia, G.Mancardi, M. Bove, and G. Brichetto. *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.

[26] `http://icaps-conference.org/index.php/Main/Competitions/`

[27] Drew Mcdermott, Malik Ghallab, Adele Howe, et al., *PDDL - The Planning Domain Definition Language*, 1998.

[28] Gaetano Tieri, Giovanni Morone, Stefano Paolucci & Marco Iosa *Virtual reality in cognitive and motor rehabilitation: facts, fiction and fallacies*, Expert Review of Medical Devices, 15:2, 107-117, 2018.