

# UNIVERSITA' DEGLI STUDI DI PAVIA

FACOLTA' DI INGEGNERIA  
DIPARTIMENTO DI INGEGNERIA INDUSTRIALE E DELL'INFORMAZIONE

DOTTORATO DI RICERCA IN BIOINGEGNERIA E BIOINFORMATICA  
XXVII CICLO - 2014

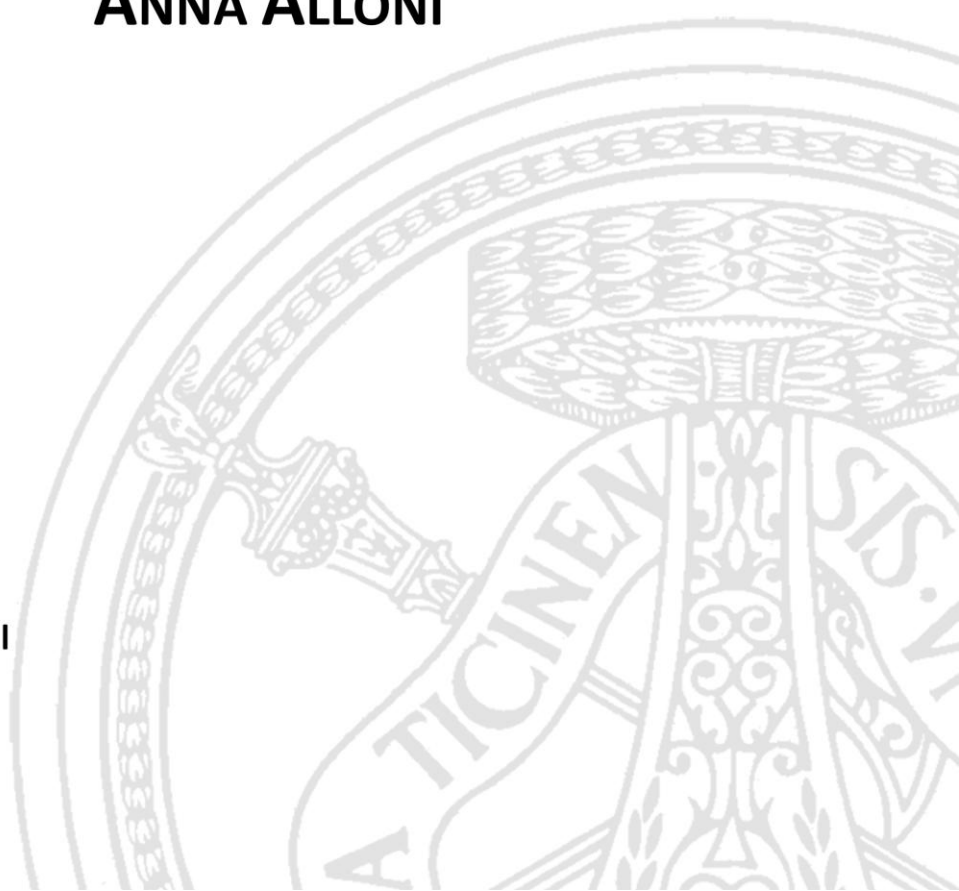
## **CoRe: A SOFTWARE SYSTEM FOR COMPUTERIZED COGNITIVE REHABILITATION**

PhD Thesis by  
**ANNA ALLONI**

**Supervisor:**  
Prof. Silvana Quaglini

**Co-supervisor:**  
Prof. Daniela Tost-Pardell

**PhD Program Chair:**  
Prof. Riccardo Bellazzi





---

## Abstract (Italiano)

---

Con il termine “processo cognitivo” si definisce l’insieme delle abilità di utilizzo ed integrazione di funzionalità cerebrali afferenti a domini cognitivi diversi:

- **percezione**, ovvero l’insieme di attività quali la consapevolezza, l’identificazione, la discriminazione, l’orientamento;
- **memoria**, cioè la capacità di richiamare gli avvenimenti del passato e generare nuove conoscenze, schemi e quadri interpretativi, fondamentali per una continua ed aggiornata valutazione del mondo;
- **attenzione**, la funzione con cui l’organismo recepisce, filtra ed organizza gli stimoli provenienti dall’ambiente esterno, allo scopo di emettere risposte adeguate. Essa è determinata da percezione, memoria e apprendimento, pensiero (che comprende funzioni cognitive come il calcolo, il ragionamento, la formazione di concetti, la valutazione, l’astrazione, la generalizzazione, la pianificazione ed il problem solving) e funzioni espressive ( parola, gestualità, espressioni facciali, movimenti);
- **linguaggio**, un sistema comunicativo costituito da singole parole, organizzate secondo regole precise per formare frasi e conversazioni;
- **azione** volontaria (l’insieme dei cosiddetti “atti motori”), che ci permette di muoverci correttamente nello spazio con finalità coerente.

Oltre all’invecchiamento fisiologico, tra le cause della perdita di tali funzionalità si annoverano patologie quali la Malattia di Parkinson e il Morbo di Alzheimer, prevalentemente associate a soggetti in età avanzata, ma anche quadri clinici quali sclerosi multipla, miocardiopatie congenite, alterazioni post-lesionali e disturbi dell’apprendimento, che spesso si manifestano anche in soggetti giovani.

Deficit cognitivi causati dall’insorgere di tali condizioni patologiche si manifestano nella vita quotidiana del paziente sottoforma di agnosie (alterazioni dell’integrazione percettiva), amnesie (deficit di memoria), afasie (alterazioni del linguaggio) e aprassie (compromissioni delle funzionalità neuromuscolari e senso-motorie) ed implicano un sensibile peggioramento della qualità di vita del soggetto.

Per contrastare il decadimento- o rallentarlo, nel caso di patologie degenerative- si ricorre alla riabilitazione cognitiva. Tale attività, ormai entrata a far parte della pratica clinica corrente, consiste nel sottoporre pazienti affetti da deficit cognitivi a batterie di esercizi pensati per

stimolare le funzionalità cerebrali compromesse, puntando a un (almeno parziale) recupero delle stesse, che si associa a un miglioramento della qualità di vita sia dell'individuo, che può riacquistare parte della propria autonomia, sia dei suoi caregivers.

La collaborazione tra l'Istituto Neurologico Fondazione Casimiro Mondino (IRCCS Mondino) e il Laboratorio di Informatica Biomedica "Mario Stefanelli" dell'Università degli Studi di Pavia ha portato alla realizzazione di CoRe, un pacchetto software che genera in forma computerizzata e personalizzabile le tipologie d'esercizio normalmente proposte in formato cartaceo e consente la visualizzazione e la rielaborazione statistica dei dati di performance acquisiti durante lo svolgimento di ogni sessione di esercizio.

Dal momento della sua creazione, il sistema è stato sottoposto a numerose modifiche, finché una versione stabile del tool è stata installata presso la Fondazione Mondino e sottoposta a una fase di test (su volontari sani) e uno studio pilota (su pazienti della struttura) i cui risultati sono qui riportati.

Questa tesi, focalizzata sullo sviluppo e l'ottimizzazione del sistema, è così organizzata:

- Il **capitolo 1** introduce il problema clinico, descrivendo brevemente le principali patologie legate all'insorgere di deficit cognitivi, così come il fenomeno della neuroplasticità, ovvero il meccanismo fisiologico di recupero che giustifica l'efficacia della riabilitazione.
- il **capitolo 2** è focalizzato sulla riabilitazione cognitiva: verrà descritto lo stato dell'arte, sia per la riabilitazione "classica" paper-based (attraverso un elenco delle principali scale di valutazione ed esercizi attualmente utilizzati nella pratica clinica), sia per l'innovativo approccio computerizzato (attraverso la descrizione di alcuni dei più noti sistemi software di riabilitazione attualmente in commercio).
- Il **capitolo 3** esamina il componente principale del nostro sistema, l'ontologia, illustrando il tipo di dominio di conoscenza necessario nel contesto della riabilitazione cognitiva, la conseguente strategia di classificazione adottata per la creazione dell'ontologia stessa e le relazioni user-defined introdotte per essere poi utilizzate per la creazione e la gestione automatizzata di esercizi relation-based. Vengono inoltre descritti gli strumenti utilizzati per costruire lo schema dell'ontologia (Protégé, un software sviluppato presso l'Università di Stanford) e tradurla in un database relazionale (un traduttore software XML-SQL realizzato ad hoc) utilizzabile dal sistema.  
Infine l'attenzione si sposta verso le capacità di personalizzazione del sistema.
- Il **capitolo 4** introduce il sistema stesso: la prima versione, basata su tool commerciali, così come la versione re-ingegnerizzata, creata per risolvere problemi legati alla prima implementazione. Oltre al sistema principale, vengono descritte anche le funzionalità aggiuntive integrate nella piattaforma.

- Il **capitolo 5** si focalizza sulla parte di sistema basata su tecnologie 3D: dopo una spiegazione delle ragioni alla base della scelta di introdurre questo modulo, vengono illustrati la tecnologia utilizzata e gli esercizi implementati.
- Il **capitolo 6** descrive la fase di validazione a cui il sistema è stato sottoposto. Il test di usabilità, effettuato su volontari sani, così come lo studio pilota su un campione di pazienti ricoverati presso la Fondazione Mondino.
- Infine, il **capitolo 7** illustra le conclusioni tratte da questa esperienza triennale: in particolare, le differenze tra l'andamento previsto del progetto e la sua effettiva evoluzione. Inoltre, vengono esaminate le possibili future direzioni di sviluppo, tra cui la proposta di progetto "Computer-based rehabilitation of communicative disorders in acquired cerebral injury", sottoposta al Ministero della Salute (programma "Ricerca Finalizzata 2013"), e HomeCoRe, l'applicazione di tele riabilitazione, descritta in tale proposta, che verrà sviluppata per i pazienti domiciliari.

---

## Abstract (English)

---

By the term "cognitive process" we define the set of abilities in use and integration of brain functions related to different cognitive domains:

- **perception**, that is the set of activities, like awareness, identification, discrimination, orientation;
- **memory**, that is the ability to recall events from the past and create new knowledge, schemas and interpretive frameworks which are essential for a continuous and constantly updated evaluation of the world;
- **attention**, the function through which the organism recognizes, filters and organizes stimuli coming from the outside world, with the goal of emitting adequate responses. It is determined by perception, memory and learning, thought (including cognitive functions as calculus, reasoning, development of concepts, evaluation, abstraction, generalization, planning and problem solving) and expressive functions (speech, gestures, facial expressions, movements);
- **language**, a communication system made up of single words, organized according to precise rules in order to create sentences and conversations;
- **voluntary action** (the set of so-called "motor actions"), that allows to correctly move in space with coherent goal.

Beside physiological ageing, other factors can be counted among the causes of the loss of such functions: pathologies like Parkinson's disease and Alzheimer's disease, which are usually associated with subjects in advanced age, but also with clinical pictures of multiple sclerosis, congenital cardiomyopathy, postlesional alterations and learning disabilities, which often arise in young subjects.

Cognitive impairments caused by the onset of these pathologic conditions affect the patient's daily life in form of agnosia (alteration of the sensory perception), amnesia (memory deficits), aphasia (speech disorder) and apraxia (impairment of neuromuscular and sensorimotor functions) and entail a sensible worsening of the subject's quality of life.

Cognitive rehabilitation is used to contrast the decay (or slow it down, in case of degenerative diseases). This activity consists in administering batteries of exercises to the patients, created to stimulate the compromised brain functions with the recovery (be it complete or partial) as the main goal. This results in an improvement of the life quality for both the patient, who regains part of his autonomy, and his caregivers.

The collaboration between the IRCCS Neurological Institute C. Mondino Foundation and the Mario Stefanelli Laboratory of Biomedical Informatics of the University of Pavia led to the creation of CoRe, a software tool that creates customizable computerized versions of usually paper-based exercises and allows the visualization and automated statistical elaboration of the performance data acquired during the execution of rehabilitation sessions. The system has undergone significant changes since its creation, and eventually a stable version has been installed at our partner healthcare facility, where it has undergone a test phase (on healthy volunteers) and a pilot study (on actual patients) whose results are reported here.

This thesis, focused on the development and optimization of said system, is organized as follows:

- **Chapter 1** introduces the clinical problem, briefly describing the principal pathologic conditions related to cognitive impairments, as well as the phenomenon called neuroplasticity, that is the physiological recovery mechanisms behind the efficacy of the rehabilitation.
- **Chapter 2** is focused on the cognitive rehabilitation: the state of the art will be explained, for both the "classic" paper-based practice (through a list of the main evaluation scales and exercises used by therapist in the clinical practice) and the innovative computerized approach (through a description of some of the most famous software systems currently available on the market).
- **Chapter 3** examines the main component of our system, the ontology, illustrating the type of knowledge domain needed in the context of cognitive rehabilitation, the subsequent classification criteria adopted in the ontology generation phase and the user-defined relations between stimuli that we chose to introduce to be exploited for the creation of relation-based exercises. Besides, the tools used to build the ontology (the Protégé software by Stanford University) and translate it into a relational database (a software XML-SQL translator) are described. Finally the focus will move to the personalization capabilities of our system.
- **Chapter 4** introduces our system: its first version, based on commercial tools, as well as the re-engineered version, created to solve problems related to the first implementation. Besides the main system, also the additional features included in the platform are explained.
- **Chapter 5** focuses mainly on the 3D-based part of CoRe: after an explanation of the reasons behind the choice to introduce this feature into the system, the technology used is illustrated and the exercises implemented are described.
- **Chapter 6** describes the validation phase undergone by the system. The usability test, carried out on healthy volunteers, as well as the

pilot study with a sample of actual patients from the Mondino Foundation.

- Finally, **chapter 7** illustrates the conclusions drawn from this three-year experience: in particular, the differences between the planned course of the phases of this work and the actual evolution of the project. Also, future work directions are examined: the project proposal "Computer-based rehabilitation of communicative disorders in acquired cerebral injury" (submitted to the healthcare ministry under the "Ricerca Finalizzata 2013" program) is described, and the characteristics of HomeCoRe, the telerehabilitation application to be developed for home patients and included in such proposal, is presented.







---

# Contents

---

<b>1</b>	<b>The clinical problem: cognitive impairment .....</b>	<b>1</b>
1.1.	<i>Etiology of cognitive impairment.....</i>	<i>1</i>
1.1.1.	Parkinson's disease .....	1
1.1.2.	Alzheimer's disease .....	3
1.1.3.	Multiple sclerosis .....	7
1.1.4.	Aphasia: types and characteristics.....	10
1.2.	<i>Neuroplasticity.....</i>	<i>14</i>
<b>2</b>	<b>Cognitive rehabilitation: state of the art.....</b>	<b>17</b>
2.1.	<i>Computerized rehabilitation.....</i>	<i>17</i>
2.1.1.	COG.I.TO.....	17
2.1.2.	Brainer.....	21
2.1.3.	CogniPlus .....	26
<b>3</b>	<b>The stimuli ontology .....</b>	<b>29</b>
3.1.	<i>The knowledge domain.....</i>	<i>29</i>
3.1.1.	Classification criteria and relations .....	30
3.2.	<i>Ontology representation.....</i>	<i>33</i>
3.2.1.	Protégé.....	33
3.2.2.	The XML structure.....	37
3.2.3.	The XML-SQL translator.....	42
3.3.	<i>Personalization.....</i>	<i>45</i>
<b>4</b>	<b>The CoRe system.....</b>	<b>47</b>
4.1.	<i>First version exploiting commercial tools.....</i>	<i>47</i>
4.1.1.	E-Prime.....	47
4.1.1.1.	E-Studio .....	48
4.1.1.1.1.	Experiment .....	49
4.1.1.1.2.	Procedure.....	49
4.1.1.1.3.	List .....	49
4.1.1.1.4.	TextDisplay .....	50
4.1.1.1.5.	ImageDisplay .....	50
4.1.1.1.6.	SoundOut.....	51
4.1.1.1.7.	Slide.....	51
4.1.1.1.8.	FeedbackDisplay .....	52
4.1.1.1.9.	Inline .....	53
4.1.1.2.	E-Run .....	53
4.1.2.	System architecture .....	54
4.1.3.	The databases .....	55
4.1.3.1.	Patient database.....	55
4.1.3.2.	Stimuli database .....	59
4.1.4.	Problems with the first version.....	60
4.2.	<i>System re-engineering.....</i>	<i>60</i>
4.2.1.	System architecture .....	61
4.2.2.	The databases .....	62
4.2.2.1.	Database personalization.....	66
4.2.2.2.	Final structure.....	67
4.2.3.	The exercises .....	68

4.3. Additional features .....	71
4.3.1. Monitoring of stress and fatigue.....	71
4.3.1.1. The device .....	72
4.3.2. Weighted score .....	75
4.3.2.1. Exercise type .....	75
4.3.2.2. Difficulty level .....	76
4.3.2.3. Execution time .....	77
4.3.2.4. Results .....	78
<b>5 CoRe3D .....</b>	<b>79</b>
5.1. Reasons for the introduction of 3D features.....	79
5.2. Background.....	80
5.3. Blender.....	82
5.3.1. Main GUI .....	83
5.4. Implementation strategy .....	85
5.5. The exercises.....	86
<b>6 CoRe validation.....</b>	<b>91</b>
6.1. Validation on healthy volunteers .....	91
6.1.1. Experimental setup .....	91
6.1.2. Results .....	94
6.2. Pilot study with a patient sample.....	102
<b>7 Conclusions and future works .....</b>	<b>110</b>
7.1. Lesson learned.....	110
7.2. Future works: design of the clinical trial "Computer-based rehabilitation of communicative disorders in acquired cerebral injury" .....	111
7.2.1. Overall Summary .....	111
7.2.2. Background / state of the art.....	111
7.2.3. Hypothesis and Significance: .....	112
7.2.4. Specific Aims: .....	112
7.2.5. Experimental Design .....	113
7.2.6. Preliminary data .....	115
7.2.7. Risk analysis, possible problems and solutions.....	115
7.2.8. Expected outcomes:.....	116
7.2.9. Significance and innovation .....	116
7.2.10. Description of the complementary and synergy research team.....	117
7.2.11. Translational relevance and impact for the National Health System (SSN).....	117
7.3. Other future works.....	117
7.3.1. HomeCoRe .....	118
7.3.1.1. Characteristics .....	118
7.3.2. Personalization .....	121
7.3.3. Ontology expansion.....	122
7.3.4. Ontology editing.....	122
7.3.5. CoRe3D.....	123
7.3.5.1. 3D models .....	123
7.3.5.2. Controls .....	124
7.3.5.3. Aids, complexity and difficulty levels.....	124
7.3.5.4. Acceptance .....	125
<b>Appendix A .....</b>	<b>127</b>
<b>Appendix B .....</b>	<b>133</b>
<b>References .....</b>	<b>156</b>
<b>List of Publications .....</b>	<b>163</b>

---

# Chapter 1

---

## The clinical problem: cognitive impairment

### 1.1. Etiology of cognitive impairment

As already mentioned, neurologic pathologies with different etiology (degenerative, vascular, traumatic, infective) lead to onset of cognitive impairments.

Knowledge of the nature and characteristics of these pathologies plays a very important role, not only in the evaluation of subjects' eligibility for the experimentation of the rehabilitation software, but also in the definition of guidelines for the realization of the system itself. For example, special peripherals like touch screens or simplified control dashboards should be integrated in a system targeted to apraxic or slightly motor-deficient patients.

For this reason, a brief explanation of some of the most important cognitive impairment-related pathologies is given below.

#### 1.1.1. Parkinson's disease

Parkinson's disease (PD) [1][2][3][4][5] is a slowly-evolving neurological pathology, due to chronic and progressive degeneration of the CNS area called *substantia nigra* (literally, "black substance", so-called because of its color, slightly darker than the surrounding areas due to high levels of neuromelanin contained), inside of which dopamine, a neurotransmitter necessary for the control of body movements) is produced (fig. 1).

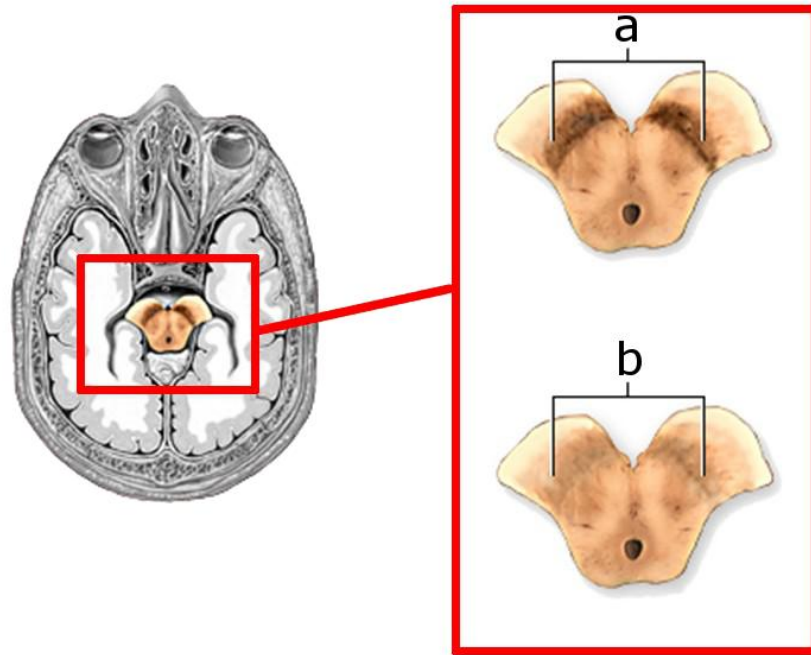


Figure 1: Location of the substantia nigra and comparison between healthy (a) and pathological (b) condition

The symptoms begin to show when the percentage of dead dopamine neurons is already dramatically high (50 to 80%): for this reason, it is very hard to study the early stages of the pathological condition.

PD is the most widespread degenerative disorder after Alzheimer's disease. Different studies show discrepancies in terms of incidence, symptoms, and response to treatments, due to variability of factors like clinical criteria adopted, enrolment procedures and composition of the reference population (in terms of age and gender).

This disease, that usually affects 55+ year old individuals, becomes more common in the 60-70 age range and shows a peak of incidence in the over-70-year-olds, though cases have been registered in young individuals, mostly due to inherited factors.

In Italy PD affects about 2% of the population aged 65 and more. According to a study by the Istituto Superiore della Sanità, the total number of cases amounts to about 220.000. [6]

The three main symptoms related to this disease are tremors, stiffness and akinesia, that is the overall decrease of voluntary and non-voluntary mobility, usually associated to slowness of movements.

Many other symptoms can become a part of a complex clinical picture, variable from patient to patient: postural alterations, subjective perception impairments, slowing down of eye movements, hypersalivation, disturbance of vegetative functions (that is, the bodily processes most directly concerned with maintenance of life: nutritional, metabolic, and endocrine functions including eating, sleeping, menstruation, bowel

function, bladder activity, and sexual performance), sleep and affective disorders.

Operational impairments are associated to this pathology: loss of concentration, slowing down of thought and inability to create linear logic connections. Also, slowness in comprehension and elaboration of concepts: this disease also entails an alteration of attention, due to a decreased ability to focus on significant stimuli (the so-called *set-shifting*)

The working memory, that is the ability to recall information already stored in mind, is also compromised, because the impairments mainly involve the mechanism of access to memories. Cognitive impairments arise in early phases of the disease and are usually detectable with a targeted neuropsychological evaluation: the mental processes related to the mechanisms of attention control and maintenance, which allow the selection of ambient stimuli and the adequate motor response, result significantly altered.

Finally, speech and writing disorders have been observed, associated to anxiety and depression symptoms.

To date, no existing drug or cure is able to prevent PD or cause it to regress: even the Deep Brain Stimulation, which represents the most advanced surgical technique currently available for PD patients, is only a temporary relief (and besides, the eligibility criteria are very restrictive, so that only 5-10% of the patient turns out to be suitable for this operation) and cannot stop the slow but constant decay.

### **1.1.2. Alzheimer's disease**

Alzheimer's disease (AD) [7][8][9] is the most common type of degenerative dementia (54% of the total reported cases), arising more frequently in population aged over 65 (with a prevalence of 4,4%). The world perspective for the year 2020 foresees more than 48 million cases of dementia (of which 15 millions only in the EU), which in the subsequent 20 years could rise up to 81 millions. [10]

For this reason the treatment of Alzheimer patients is considered a priority matter that has to be tackled with all the available instruments. Another reason why this pathology has gained attention of researchers worldwide is represented by the major implications it has on the life of affected patients: the onset of AD implies a progressive decay of cognitive functions, starting from memory. Its causes are still unknown, although numerous risk factors have been identified: old age, family history, brain traumas, lifestyle and conditions causing blood vessel problems.

The progression of the disease is characterized by cortical atrophy, distinguishable by the enlargement of cerebral *sulci* and corresponding flattening of cerebral *gyri* (fig. 2), as well as by presence of amyloid plaques and neurofibrillary tangles into the brain tissue (fig. 3).

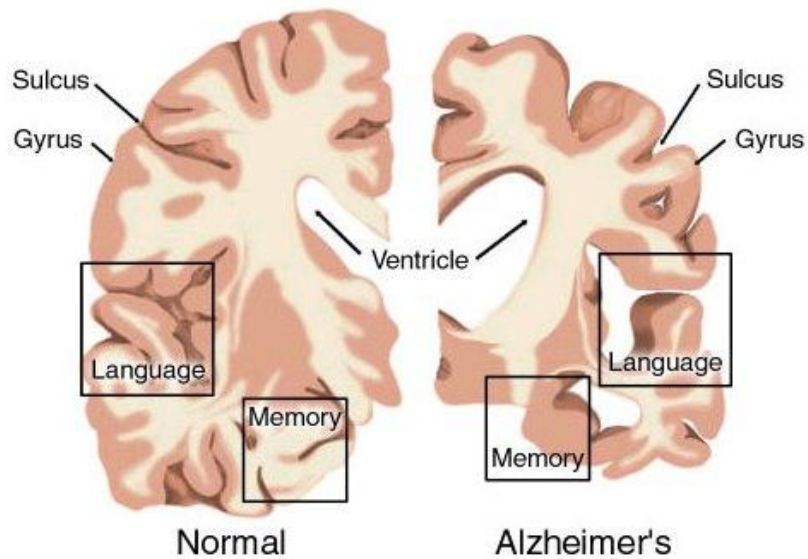


Figure 2: Comparison between a healthy brain (left) and one with Alzheimer's disease (right). The deterioration of areas where the functions of memory and language are located is clear.

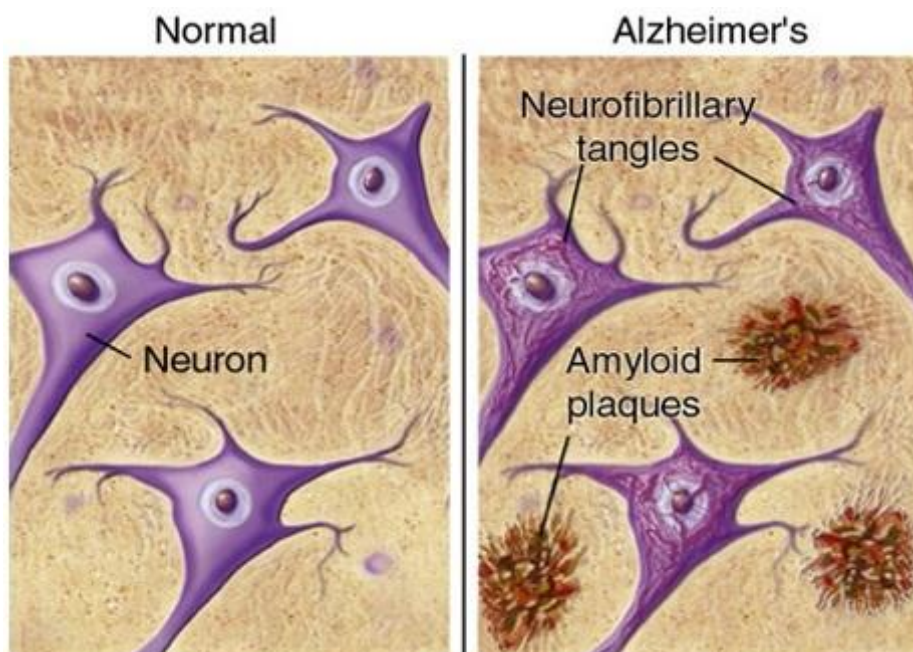


Figure 3: Appearance of the cerebral tissue in normal (left) and pathologic (right) condition

Amyloid plaques are deposits of beta-amyloid peptides caused by processing of Amyloid Precursor Protein (APP). In a healthy brain, they are disrupted and disposed of, but in a diseased one they collect and form hard layers which are not soluble. Despite lacking absolute certainty of amyloid being a cause of Alzheimer's disease (AD), abnormalities in its presence in



the brain are considered a reliable indicator for an early onset of the disease.

Neurofibrillary tangles are twisted fibers of Tau protein, an element that can be found in a healthy brain (Tau protein is a component of the microtubules that transport nutrients between areas of the brain). Hyperphosphorilation of this protein causes its aggregation in tangled masses, compromising the stability of the microtubules and causing the collapse of the transport system, as well as the progressive shrinkage of the hippocampus.

From a functional point of view, the above described nerve degeneration translates to memory loss, which is the first sign of AD onset.

Evaluation of Alzheimer patients is performed following the so called Global Deterioration Scale, elaborated by dr. Barry Reisberg [11], reported below, in table 1. The progression of the dementia is divided in 7 stages, escalating from the condition of normality to cognitive impairment to severe dementia. The description of the typical symptoms characterizing each stage is a clear indicator of the depth of the impact this disease has on a patient's life.

Table 1: Global Deterioration Scale for assessment of dementia

Level	Clinical Characteristics
1 No cognitive decline	No subjective complaints of memory deficit. No memory deficit evident on clinical interview.
2 Very mild cognitive decline (Age Associated Memory Impairment)	Subjective complaints of memory deficit, most frequently in following areas: (a) forgetting where one has placed familiar objects; (b) forgetting names one formerly knew well. No objective evidence of memory deficit on clinical interview. No objective deficits in employment or social situations. Appropriate concern with respect to symptomatology
3 Mild cognitive decline (Mild Cognitive Impairment)	Earliest clear-cut deficits. Manifestations in more than one of the following areas: (a) patient may have gotten lost when traveling to an unfamiliar location; (b) coworkers become aware of patient's relatively poor performance; (c) word and name finding deficit becomes evident to intimates; (d) patient may read a passage or a book and retain relatively little material; (e) patient may demonstrate decreased facility in remembering names upon introduction to new people; (f) patient may have lost or misplaced an object of value;

The clinical problem: cognitive impairment

	<p>(g) concentration deficit may be evident on clinical testing. Objective evidence of memory deficit obtained only with an intensive interview. Decreased performance in demanding employment and social settings. Denial begins to become manifest in patient.</p> <p>Mild to moderate anxiety accompanies symptoms.</p>
<p>4 Moderate cognitive decline (Mild Dementia)</p>	<p>Clear-cut deficit on careful clinical interview. Deficit manifest in following areas: (a) decreased knowledge of current and recent events; (b) may exhibit some deficit in memory of ones personal history; (c) concentration deficit elicited on serial subtractions; (d) decreased ability to travel, handle finances, etc.</p> <p>Frequently no deficit in following areas: (a) orientation to time and place; (b) recognition of familiar persons and faces; (c) ability to travel to familiar locations. Inability to perform complex tasks. Denial is dominant defense mechanism. Flattening of affect and withdrawal from challenging situations frequently occur.</p>
<p>5 Moderately severe cognitive decline (Moderate Dementia)</p>	<p>Patient can no longer survive without some assistance. Patient is unable during interview to recall a major relevant aspect of their current lives, e.g., an address or telephone number of many years, the names of close family members (such as grandchildren), the name of the high school or college from which they graduated. Frequently some disorientation to time (date, day of week, season, etc.) or to place. An educated person may have difficulty counting back from 40 by 4s or from 20 by 2s. Persons at this stage retain knowledge of many major facts regarding themselves and others. They invariably know their own names and generally know their spouses' and children's names. They require no assistance with toileting and eating, but may have some difficulty choosing the proper clothing to wear.</p>
<p>6 Severe cognitive decline (Moderately Severe Dementia)</p>	<p>May occasionally forget the name of the spouse upon whom they are entirely dependent for survival. Will be largely unaware of all recent events and experiences in their lives. Retain some knowledge of their past lives but this is very sketchy. Generally unaware of their surroundings, the year, the season, etc.</p> <p>May have difficulty counting from 10, both backward and, sometimes, forward.</p>

	<p>Will require some assistance with activities of daily living, e.g., may become incontinent, will require travel assistance but occasionally will be able to travel to familiar locations. Diurnal rhythm frequently disturbed. Almost always recall their own name. Frequently continue to be able to distinguish familiar from unfamiliar persons in their environment. Personality and emotional changes occur. These are quite variable and include: (a) delusional behavior, e.g., patients may accuse their spouse of being an impostor, may talk to imaginary figures in the environment, or to their own reflection in the mirror; (b) obsessive symptoms, e.g., person may continually repeat simple cleaning activities; (c) anxiety symptoms, agitation, and even previously nonexistent violent behavior may occur; (d) cognitive abulia, i.e., loss of willpower because an individual cannot carry a thought long enough to determine a purposeful course of action</p>
<p>7 Very severe cognitive decline (Severe Dementia)</p>	<p>All verbal abilities are lost over the course of this stage. Frequently there is no speech at all -only unintelligible utterances and rare emergence of seemingly forgotten words and phrases. Incontinent of urine, requires assistance toileting and feeding. Basic psychomotor skills, e.g., ability to walk, are lost with the progression of this stage. The brain appears to no longer be able to tell the body what to do. Generalized rigidity and developmental neurologic reflexes are frequently present.</p>

Following this classification, at stage 5 the subject is not able to live an independent life anymore, and must rely on constant assistance by a caregiver, whose role becomes more and more important, until the point when the psychomotor skills are completely compromised and the patient is no longer able to undergo activities which are basic for survival.

To date, no cure exists for Alzheimer's disease: available treatments only alleviate the symptoms and, in some cases, slow the disease progression.

### 1.1.3. Multiple sclerosis

Multiple sclerosis (MS)[13][14][15][16] is an autoimmune inflammatory disorder that affects the brain and spinal cord. Cells of the immune system (T-lymphocytes, or T-cells) are able to cross the vessel walls and penetrate the Central Nervous System (CNS) by infiltrating the hematoencephalic

barrier (also called Blood-Brain Barrier, BBB), a structure of capillary endothelial cells connected by tight junctions that acts as a highly selective permeability barrier separating the circulating blood from the brain extracellular fluid.

In normal conditions, the CNS would be inaccessible to white cells, but magnetic resonance imaging shows that during a MS attack the BBB has broken down, thus allowing the crossing of the T-cells, which proceed to attack the myelin sheath covering the axons (fig. 4).

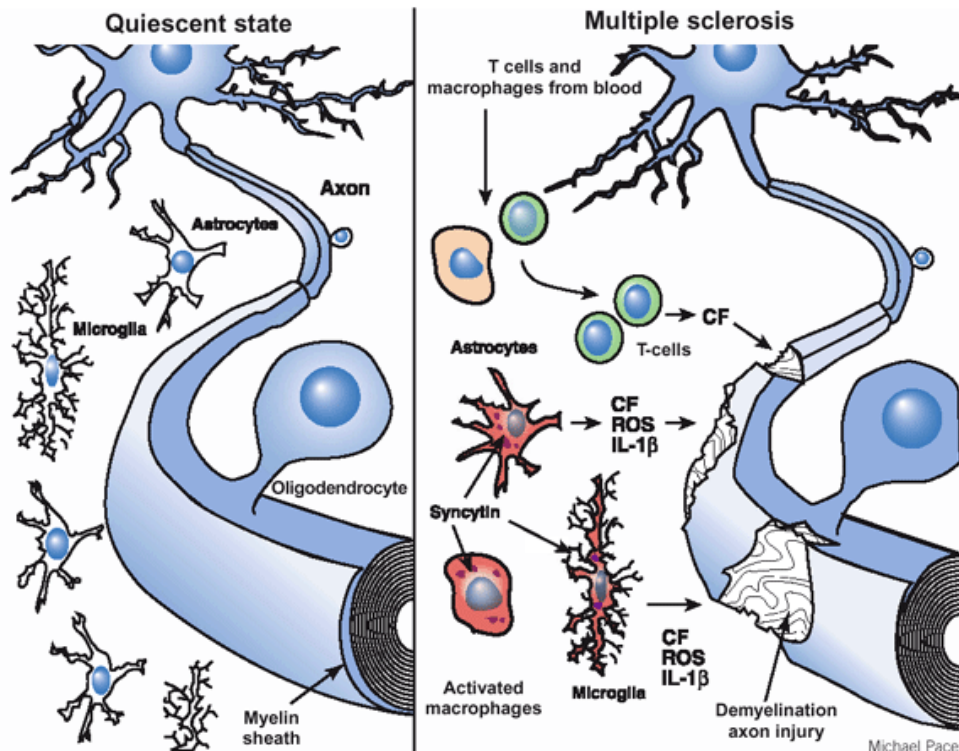


Figure 4: Comparison between an axon in quiescent state and one during a MS "attack"

Destruction of myelin implies impairment of nerve impulse transmission: for this reason, the resulting symptoms are determined by the function of the involved nerves and affect, among others, motor, sensory, visual and autonomic system. Evoked potentials can be useful for the diagnosis: detection of a prolonged latency can be interpreted as the effect of demyelination on saltatory conduction.

There are several events which can be considered indicators of MS insurgence: first of all, white matter abnormalities, though not enough to infer a diagnosis alone (many people aged over 50 show white matter cerebral lesions), are present in more than 95% of MS patient.

In the initial phase the remyelination of the damaged axons can occur, so the inflammation and consequent neurologic dysfunction is only temporary;

in time, though, the recovery is incomplete: the neurodegeneration accumulates, and so does the disability, becoming chronic and extensive.

About half the MS subjects develop cognitive impairments, especially related to certain brain functions:

- Memory
- Attention and concentration
- Information processing
- Executive functions
- Visuospatial functions
- Verbal fluency

On the contrary, some functions are often left intact:

- General intellect
- Long-term memory
- Conversation ability
- Reading comprehension

Only 5 - 10% of subjects show impairments so serious that the disease interferes with everyday life; the cases for which the severity is such that homecare is not a viable option are even more rare.

The clinical course of the illness usually lasts several decades: the median death time is about 30 years from the onset.

In general, the characteristics of MS-related cognitive impairments are the following:

- They can occur at any time, but are more common in later phases of the disease;
- They generally progress slowly but are unlikely to improve dramatically once they have begun;
- Cognitive function correlates with number of lesions and lesion area on MRI, as well as brain atrophy.

A careful evaluation is necessary to determine the cause(s) of mental changes since cognitive function can also be affected by aging or medications, as well as depression, anxiety, stress and fatigue.

1. A specially trained health professional (neuropsychologist, speech/language pathologist, or occupational therapist) administers a battery of tests to evaluate cognitive dysfunction
2. Based on the test findings -including the person's cognitive deficits and strengths- the health professional can provide cognitive rehabilitation, including computer-mediated memory exercises and training in the use of compensatory strategies such as notebooks, computers and filing systems to compensate for memory problems and other changes.

Although no treatments exist that can heal the disease, several methods can be applied to mask individual symptoms, with the goal of improving the patient's everyday life by reducing the impact of the impairment. For subjects in advanced stage, with serious disability, comprehensive care includes access to occupational therapists, neuropsychologists and, in general, professionals able to provide help with the management of chronic neurologic disease.

#### **1.1.4. Aphasia: types and characteristics**

Aphasia [17][18][19][20][21] is a cognitive disorder caused by deficits in brain areas dedicated to language-related functions. It is caused by acquired damage - most commonly due to head injuries and stroke, but also cancer, epilepsy and Alzheimer's disease (acute aphasia), or even brain tumors or dementia (progressive aphasia). The range of disorders entailed by this condition is very wide and goes from difficult recall of words to loss of ability to understand, speak, read and/or write.

The clinical picture associated to this disease is very complex, and the related disorders have a huge impact on a subject's life, even if this condition appears less severe and damaging than the diseases described above. To give an idea of what aphasia feels like for a patient, this is how the site of the Ontario Association of Speech-Language Pathologists and Audiologist describes it:

*"imagine yourself in a foreign country where you don't speak the language, can't read a newspaper, or need to write with a different alphabet. You would have a lot of trouble understanding, but you certainly wouldn't think that you had suddenly lost your intelligence" [22]*

Based on the severity of the aphasic condition and the brain function it affects, the patient can find himself in very different situations: it might be hard for him to understand/read/write/speak even the simplest words, or (if the condition is particularly mild) an observer may not even notice that a problem exists.

In order to make some clarity on the effective width of the range of medical conditions implied by this disease, a detailed and thorough description of the types of aphasia (and the relevant classification criteria) is provided below.

Neuropsychology offers several interpretations for the classification of the aphasic picture, a topic which has been the center of debate since the XIX century. Paul Broca, the french neurologist after whom the namesake type of aphasia has been called, hypotesized that speech production happens in the left brain hemisphere, and upon studying the well-known case of his patient Tan (lack of ability to produce cogent speech associated to intact speech production and other brain functions following a cerebral lesion located in the left hemispere) discovered that his disease was the

consequence of "the loss of a particular type of memory, not the words memory, but the one of the movements necessary to express them". [23]

Carl Wernicke was the first to produce a linguistic model. He identified two areas of the brain, the *Auditory-Verbal Centre* (AVC) and the *Verbal-Motor Centre* (VMC), connected to each other by a bundle of axons called the *Arcuate fasciculus* (AF) and cooperating for the representation of words (fig. 5).

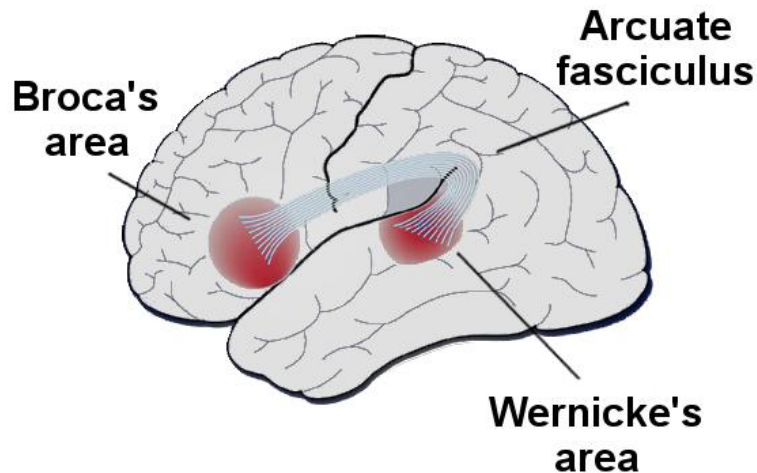


Figure 5: Location of Broca's area, Wernicke's area and the Arcuate fasciculus

Wernicke proposed a diagram for the elaboration of oral and written language, in which the written production depended on the oral one - specifically, on the AVC. Based on this model, he proposed a systematic classification of different forms of aphasia:

- sensory aphasia (Wernicke's Aphasia), a consequence of the destruction of the AVC, that manifests itself in comprehension and repetition impairments coupled with unawareness of the paraphasia shown.
- conduction aphasia, due to the destruction of the association fibers between AVC and VMC. It is characterized by adequate comprehension, deficit in repetition and articulation, and deficit awareness by the patient.
- motor aphasia (Broca's aphasia), characterized by severe articulatory deficits following a lesion of the VMC, normal comprehension ability and occasional deficits affecting written language.

To bridge the gaps emerged in Wernicke's model, which was considered non exhaustive (since it was unable to explain some clinical pictures),

Ludwig Lichtheim created a more complex schematization, in which the so-called Concept Centre (CC) was introduced. This model allowed the explanation of reading and writing disorders by hypothesizing the existence of specific brain centres (fig. 6).

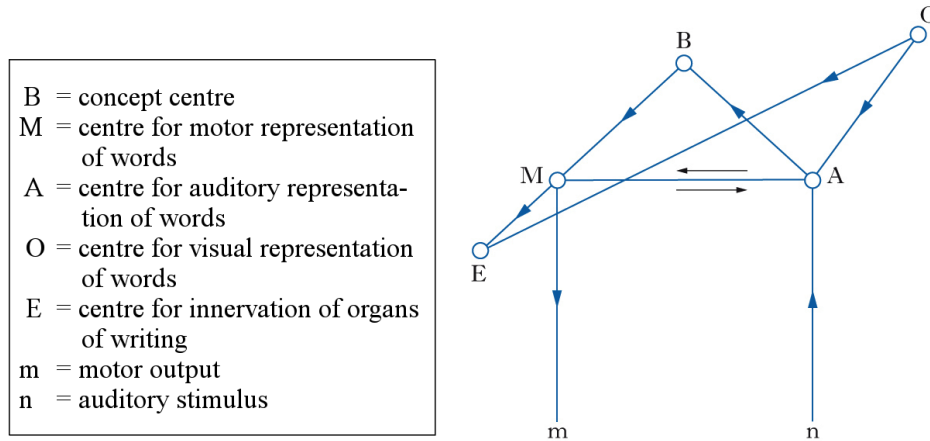


Figure 6: Lichtheim's language model

Each of the centres cooperates with others for the execution of tasks: for example, writing under dictation corresponds to the n-A-M-E path, while comprehension of written words involve the O-A-B centres.

Lesions affecting the represented paths correspond to selective behavioral deficits (and thus, to different types of aphasia):

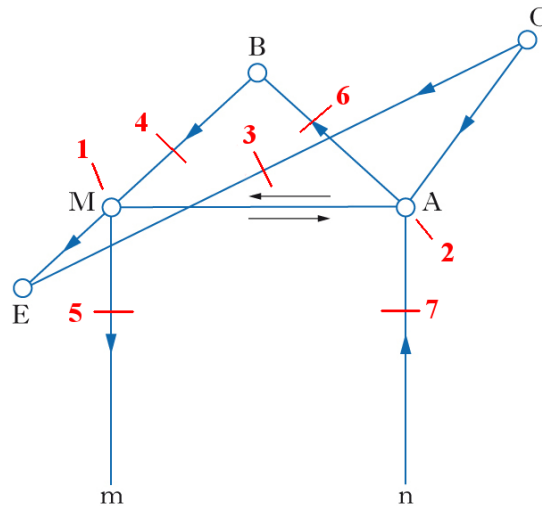


Figure 7: Lesions represented as path disruptions in the Lichtheim language model



- 1 causes motor aphasia (Broca's aphasia). Tasks in which the M centre is involved are compromised: spontaneous speech (BMm), repetition of words (nAMm), reading aloud (OABMm).
- 2 causes sensory aphasia (Wernicke's aphasia). The lesion affects the centre for auditory representation of words, so comprehension of oral (nAB) and written language (OAB), as well as repetition of words (nAMm), writing under dictation (nAME) and reading aloud (OBAMm) are compromised.
- 3 causes conduction aphasia and affects all the tasks where hearing and articulation are involved (repetition, reading aloud etc.)
- 4 causes motor transcortical aphasia
- 5 causes motor subcortical aphasia
- 6 causes transcortical sensory aphasia
- 7 causes subcortical sensory aphasia

Not only Lichtheim's model was successful in establishing a direct correspondence between impaired brain functions and selective lesions of functional components, thus giving an explanation for specific aphasic states, but it also became the reference schema for some of the most important batteries for evaluation of aphasia created in the 1970s and 1980s, like the Boston Diagnostic Aphasia Examination (BDAE), the Western Aphasia Battery (WAB) and the Aachener Aphasie Test (AAT).

Another classification model exists, created by Harold Goodglass in 1994, that distinguishes aphasia types in two big categories, fluent and non fluent, under which all the known types of aphasia can be classified depending on the oral verbal production of the patients.

The algorithm for the diagnosis of aphasic syndromes is shown below:

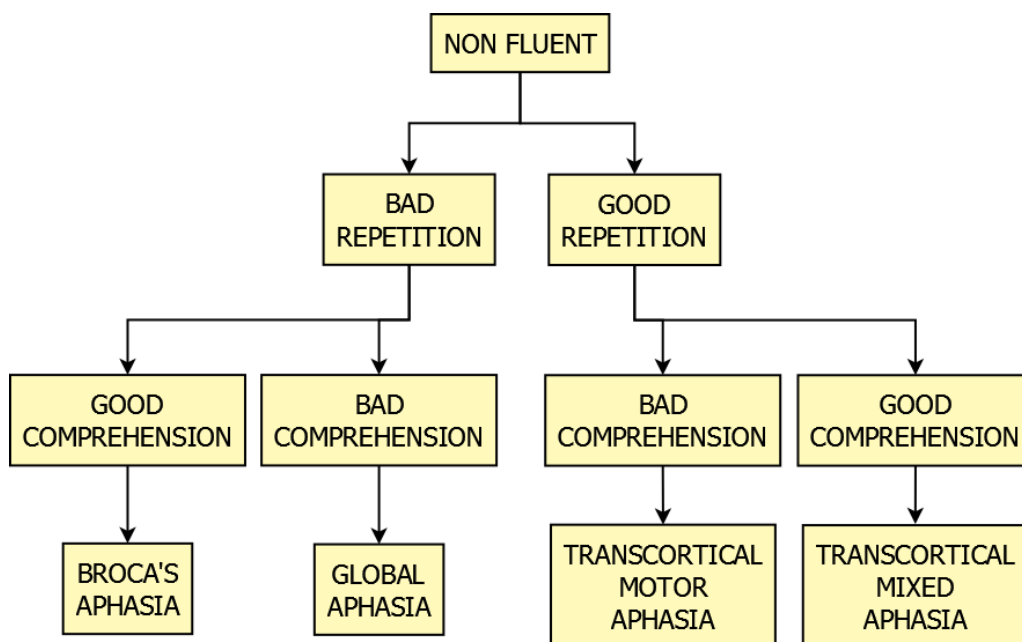


Figure 8: Classification of non fluent aphasia

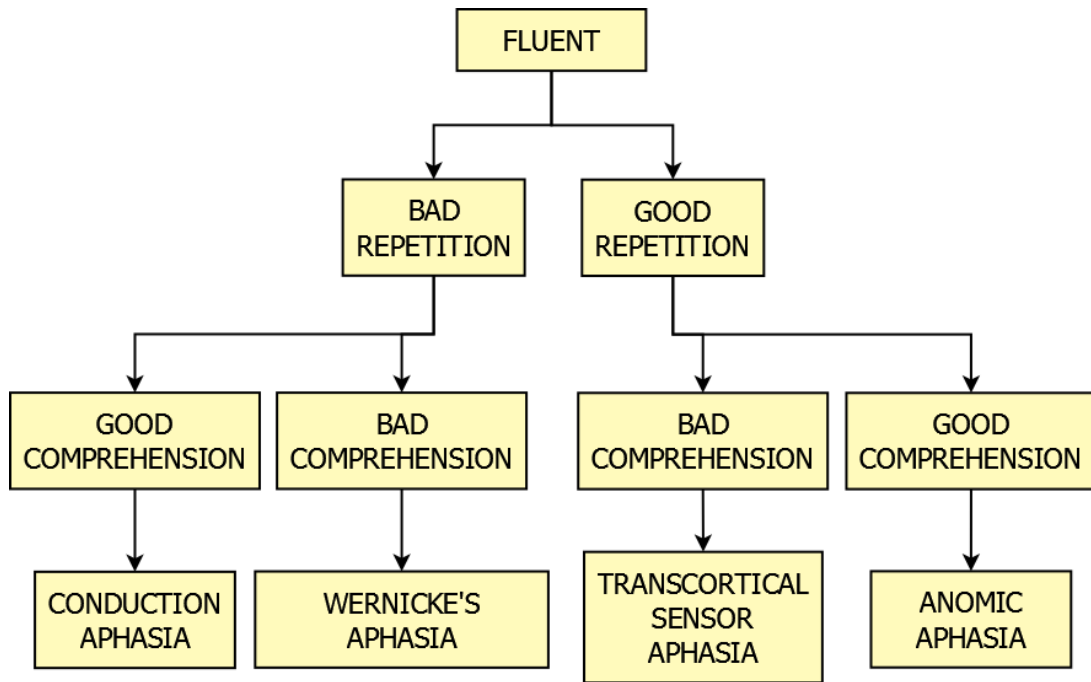


Figure 9: Classification of fluent aphasia

## 1.2. Neuroplasticity

Logical-executive cognitive functions are the ones most compromised by the above-described neurologic pathologies. They represent a complex system of brain functional modules where the regulation of the cognitive system's planning, control and coordination processes reside.

These functions are located into the frontal lobe- precisely, in the pre-frontal cortex, which is thought responsible for the functional organization of the higher order thinking.

This area represents the most important part of a wide network of nerves with a high number of afferences and efferences, whose interconnections can reach every area of the CNS and are fundamental for activation, inhibition and integration of ideomotor and sensimotor activity, as well as for the organization of relational, social and environmental behavior.

The rehabilitative approach is based on a defined premise, called brain plasticity [24][25]. This term indicates the nervous system's ability to develop structural and functional modifications as an answer to different types of events:

- physiological events, usually occurring during the development of the encephalon, the moment when the plasticity is at its peak;
- environmental stimuli, thanks to which the learning process occur;
- pathologic events, entailing lesions of the nervous system (in this case brain plasticity represents a spontaneous recovery mechanism).

Of the various mechanisms through which brain plasticity manifests itself, synaptic plasticity is worth mentioning. It is the ability to modify the structures assigned to the propagation of nervous signals (synapses) to facilitate the creation of new connections and/or the rearrangement those already strong, depending on the functional activity. In processes like memory and learning, the adaptability of the nervous system to different environmental conditions is expressed with the promotion or inhibition of the nervous impulse. It is brain plasticity, on which these processes are based, that makes functional reorganization and structural rearrangement of neural circuits possible.

Functional reorganization allows the recovery of compromised abilities thanks to intact neural structures which take charge of functions normally demanded to the damaged parts of the neural network, also maintaining the ability to carry out their own activities.

In case of structural rearrangement, the recovery of the function is made possible by a process of anatomic reconstruction taking place at the damaged areas, where the connections are broken.

In both cases, the intact nerve cells involved in the regeneration process are the ones located in areas topographically adjacent to the injured part.

Following a lesion, neurons of the injured area are "muted" and unable to receive stimuli. A functional NMR shows how, in a matter of hours, they start to answer stimulations by regions adjacent to the lesion, which appear to have replaced the damaged network in carrying out their tasks.

According to recent literature, brain plasticity can be observed in lesions occurring to both developing and adult brains; besides, plastic activity has been recognized within the CNS of adult individuals (while in the past it was attributed to the peripheral nervous system only). This corroborates the hypothesis of plasticity lasting until advanced age. Experimental research has been carried on, both on primates and humans, that provides evidence of massive cortical rearrangement as a consequence of either sensory input suppression in adult age or cortical disruptions (both spontaneous and induced). Studies also show how post-lesional training can restore plasticity and functional recovery.

By stimulating the brain in the correct way it is possible to compensate lesions, favoring a neuroplastic response. It is necessary to observe that within the context of structural and functional rearrangement, interfering abnormal activities can arise, which need to be inhibited in order to facilitate the post-lesional recovery.

If on the one hand it is clear how training and the correct execution of tasks modify the nervous system with the aim of optimizing the function, on the other it is true that continuous and improper exercise can cause an unwanted effect of disorganization of the neural modifications.

Besides, also the environmental context of the rehabilitation comes into play: after a brain injury, structural modifications and reorganization of functions happen correctly only in presence of an environment that guarantees adequate stimulations, aimed at compensating motor, sensor and cognitive deficits caused by the lesion itself.



---

# Chapter 2

---

## Cognitive rehabilitation: state of the art

Though, as mentioned, a cure does not exist for the above described pathologies, which are irreversibly degenerative, studies show that keeping the brain active and stimulating the synapses is a way to slow down the progression of the decay and to maintain the patient's independence as much as possible.

Cognitive rehabilitation [26][27][28][29][30][31] has two main goals: first of all, the achievement of the maximum possible functional recovery, and second, the improvement of the patient's subjective well-being, to be obtained by facilitating his reintegration into the familiar and social environment.

In general, rehabilitation aims at preventing secondary complications, curing and reducing residual disabilities and neurologic deficits and maintaining residual functions.

The study of techniques of cognitive rehabilitation originates from the attempt to provide models for recovery of neurological and neuropsychological functions, altered by brain pathology.

### 2.1. Computerized rehabilitation

Several tools for computerized cognitive rehabilitation are available on the market, each one with interesting characteristics and features that offer therapist and patients a valuable support throughout the recovery path. The most important are briefly described below.

#### 2.1.1. COG.I.TO.

COGITO [32] is a computerized tool aimed at the rehabilitation of adults afflicted by neurologic pathologies, attentive deficits, visuospatial memory problems and language impairments.

It's a free open-source web-based application that features an instruction manual to guide users through the installation and use.

The home page lists the available categories, together with the links for the visualization of the results obtained by the selected patient.

Entering an exercise page within a category, execution instructions appear on screen, together with a screenshot of the actual exercise, to help the patient understand the type of interaction required for the resolution.

The categories "selective attention" and "visuospatial memory" feature interesting image-based exercises, while "language" contains about a dozen different text-based tasks. The most significant are listed below:

- Oral comprehension of words: an auditory stimulus is played (a voice pronouncing a word); the patient must select the image that corresponds to that term.
- Written comprehension of words: upon reading a word, the patient must select the the image that corresponds to that term (fig. 10);

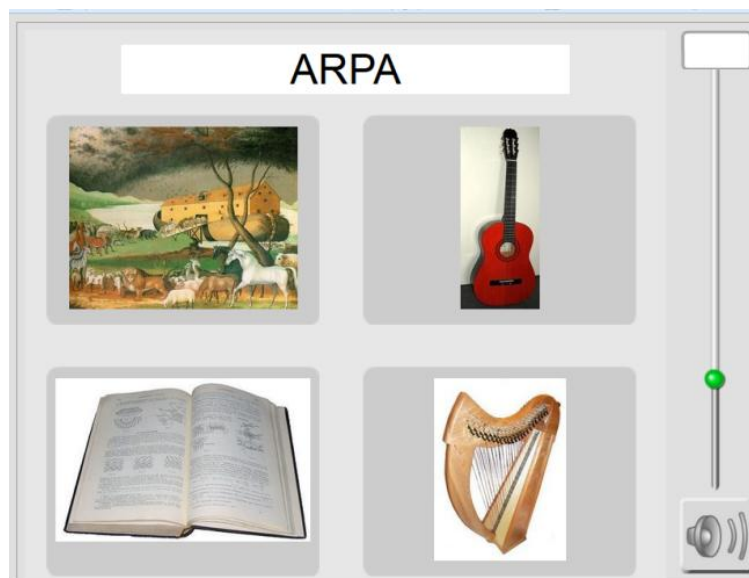


Figure 10: Instance of "Written comprehension of words"

- Oral comprehension of sentences: a voice describes a scene; the patient must select the the image that shows that scene;
- Written comprehension of sentences: upon reading a sentence describing a scene, the patient must select the image that shows that scene (fig. 11);



Figure 11: Instance of "Written comprehension of sentences"

- Sentence unscrambling: after examining an image on screen, the patient must reconstruct the sentence that describes the picture (Fig. 12).



Figure 12: Instance of "Sentence unscrambling"

Other language tasks are available, but cannot be executed without the assistance of the therapist, since the automation is not complete.

Within the "selective attention" category, two exercises are available: "prato fiorito" and "tavolo con bottoni", which are two visual variations of the same task.

They show a background image (a lawn or a tablecloth) over which some objects are spread, of different shapes and colors (flowers or buttons, see fig. 13).



Figure 13: Instance of "Prato fiorito". The task requires the selection of yellow flowers

The user must pick the object types requested by the system. The setup stage requires definition of several essential parameters; more difficulty levels are available, determining the complexity of the scene (number of distractors and targets). The user can also choose the selection criterion (by color, by shape, by color AND shape etc).

The exercise starts with a screen showing the instructions ("find all the yellow flowers", "find all the square buttons" etc). After that, the execution starts; the sidebar shown during the execution offers some visual facilitators: a grid, to scan the scene with more precision, a "container" where all the picked objects are stored; a black strip to hide part of the scene and allow the user to focus on a narrower portion of the screen.

Once the task is completed, a feedback is presented, with charts providing an insight on the results (number of false positives, false negatives etc)

"visuospatial memory" features 3 types of tasks: repositioning, recognition, memory. All of them are based on placement of specific objects within a meaningful background (kitchen, bathroom, bedroom, garage).



Repositioning is a memory task: at first the background complete with all the objects is visualized. After some time the objects disappear and the user must place them in their original position. The exercise is only over when all of the objects are put back to their place.

Recognition is based on the same principle: this time not all of the objects proposed by the system must be placed into the scene: some are intruders, automatically introduced as distractors by the system.

Memory is basically the same task as repositioning, but the user is also required to identify the objects by name (fig. 14): clicking on the original position of an object, it is possible to type the name through the keyboard. If the name matches the one required by the exercise, the object is considered identified.



Figure 14: Instance of "memory". Clicking on the pencil the keyboard input is activated

The software allows the creation of statistics concerning the entire rehabilitation path of the patient, the current sitting and the specific exercise type.

### 2.1.2. Brainer

From the home page of the Brainer site [33], it is possible to access all the available exercises. Just by looking at the interface, offering to select one of the 8 possible brain functions to be rehabilitated, it is immediately noticeable how wide the spectrum of the software is (fig. 15).



Figure 15: the main interface of Brainer

All the available exercises offer more difficulty levels that can be manually adjusted by the therapist during the setup of the sessions. Beside types already seen in COGITO, based on visual and auditory perception of text/image stimuli, Brainer has also available exercises concerning:

- Attention: a series of text stimuli is visualized on screen; one of them disappears and the user must identify it among the options proposed;



Figure 16: Instance of a digit-based attention exercise

language exercises requesting, for example, to identify, among the ones proposed, the word starting with a different syllable than the others (fig. 17)



Figure 17: Instance of one of the language exercises: the words are SACCO, SASSO and TACCO. This particular instance is not very accurate, considered that none of the three words starts with the same syllable (SAC-SAS-TAC)

- Calculation: counting with the abacus. A number is visualized and the user must move the spheres of an abacus to match that quantity



Figure 18: Instance of "Counting with the abacus"

- Calculation: sums. The user must choose among the available options the correct result of the sum proposed.
- Logic and recreation: rebuild the complex figure. A figure composed by different modules is shown. The patient must choose the right components and combine them to recreate that figure (see below).



Figure 19: Instance of "Rebuild the figure"

- Logic and deduction: select the non-pertinent image. The user must pick the intruder within a set of pictures.



Figure 20: Instance of "Select the non-pertinent image". Particular care must be put in the construction of relation-based exercises like this one, in order to avoid potential misinterpretations. In this particular instance more than one image can be considered an intruder, based on the criterion chosen.

- Memory: a sequence of visual and auditory stimuli is played. The user must repeat it without mistakes.
- Memory: match word to image. A text appears on screen for some seconds. Then it's replaced by two images. The user must select the one that represents the word.

From the point of view of the statistics elaboration, it is possible to visualize charts of the performance parameters (fig. 21). By clicking on the single series points, it is possible to get more detailed information regarding the specific cognitive functions.

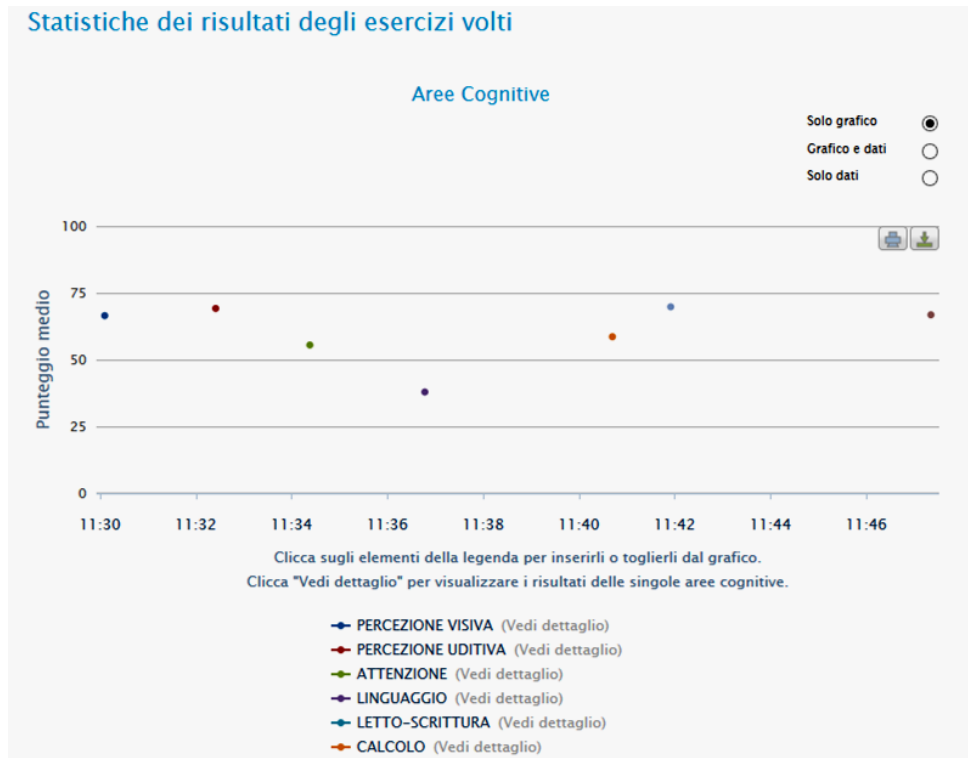


Figure 21: Example of the charted performance data. Different colors of the points represent different cognitive areas (see the legend)

An interesting feature of the Brainer software is the weighted score, a value expressed in hundredths (basically a percentage) which, as it can be deduced by looking at the table shown in the image below, is based on

- exercise type;
- accuracy;
- response time;
- difficulty level.

The main purpose behind the introduction of the weighted score is the necessity to be able to confront performances obtained in different exercises, values which otherwise wouldn't be comparable.

Report del 19.04.2013  
Totale giochi eseguiti: 13

Data di esecuzione	Area cognitiva/Nome del gioco	Risposte corrette/Risposte totali	Tempo impiegato/Tempo totale (sec.)	Livello	Punteggio ponderato
2013-04-19 11:34:23	ATTENZIONE / Individua il numero scomparso all'interno della sequenza proposta	4 / 5	35.3 / 180	Facile	49.9
2013-04-19 11:35:27	ATTENZIONE / Individua il numero scomparso all'interno della sequenza proposta	4 / 5	31.9 / 180	Medio	61.1
2013-04-19 11:40:42	CALCOLO / Somma	4 / 5	15.7 / 180	Facile	58.6
2013-04-19 11:36:47	LINGUAGGIO / Individua tra le immagini la sillaba iniziale diversa	2 / 5	31.3 / 180	Facile	37.9
2013-04-19 11:43:38	LOGICA E DEDUZIONE / Individua la figura non pertinente	5 / 5	14.5 / 180	Facile	60.3
2013-04-19 11:43:53	LOGICA E DEDUZIONE / Individua la figura non pertinente	5 / 5	9.9 / 180	Medio	83.4
2013-04-19 11:41:55	LOGICA E DEDUZIONE / Ricrea la figura composta	5 / 5	34.7 / 180	Facile	65.5
2013-04-19 11:47:21	MEMORIA / Memorizza e abbina la parola alla figura corrispondente	5 / 5	12.7 / 180	Facile	68.4
2013-04-19 11:47:42	MEMORIA / Memorizza e abbina la parola alla figura corrispondente	5 / 5	4.7 / 180	Medio	84.3

Figure 22: End-session report showing the weighted score obtained for each exercise.

### 2.1.3. CogniPlus

Widely known on an international level, CogniPlus [34] is the evolution of RehaCom.

The structure of its training sets aims at reproducing real-life situations as best as possible, under the hypothesis that this favors the patient's reintegration in the social environment. This can be considered as the system's main strong point, and the characteristic that

CogniPlus's areas of interventions are:

- Attention
- Field of view
- Working memory
- Long term memory
- Executive functions
- Spatial elaboration
- Visuomotor coordination

Numerous exercises are available within the system (fig.23), organized in functional test sets [35] aimed at rehabilitation of one of the cognitive areas listed above. In general, great care has been put into keeping the functioning mechanism simple: many exercises request the user to just

press a button to react to specific events. Based on the type of the request, the same mechanism can be adapted to the rehabilitation of different functions.

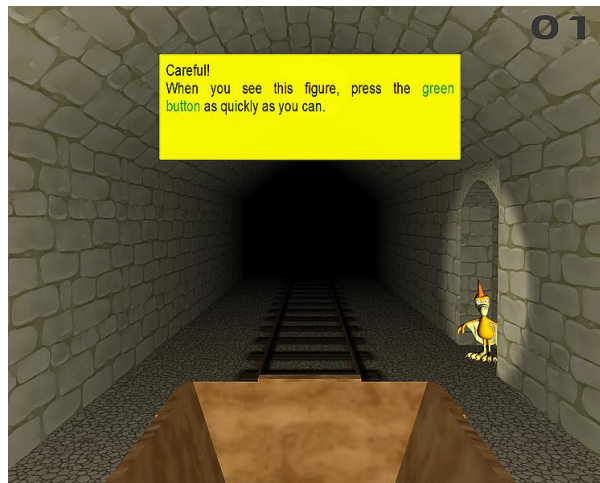


Figure 23: Examples of exercises available in CogniPlus.

Another important characteristic of CogniPlus is the automatic adaptability to the patient's abilities, that makes it possible to use the same tool throughout the whole rehabilitation process without having to manually set it up whenever the patient's status changes.



---

# Chapter 3

---

## The stimuli ontology

The innovative methodological approach chosen to implement our system (and also, the characteristic that differentiates it from all the other rehabilitation software solutions currently on the market) consists in using an ontology to store and classify all the stimuli available in our repository.

The ontology [36][37][38][39] is the ideal framework for the description of knowledge. It allows the representation of both abstract and concrete concepts (also called classes or types) that are needed to be used in a specific domain. Attributes can be associated to every concept, allowing the storage of useful data representing specific characteristics of every single entity.

Concepts are not isolated: they are actually interconnected through relations, another type of elements also contained in the ontology; the default one, "*is\_a*", on which the creation of the hierarchical structure is based, implies that one of the two entities linked (the *child* concept) can inherit properties and relations from the other, that is its *parent* class. Beside this *subsumption* relation, fundamental to structure a taxonomy, many other types of connections can be introduced into the schema by the user, to allow the creation of a more complex and satisfactory model of reality- which is exactly the goal an ontology aims at.

### 3.1. The knowledge domain

The challenging aspect in the implementation of an ontology within the field of cognitive rehabilitation is represented by the width of the domain to be modeled. Contrary to what happens for the majority of the existing ontologies, which typically are focused on representing a specific and clearly bounded field of the cognizable reality, in the field of cognitive rehabilitation the greater the number and *variety* of available stimuli, the better. That's why, as we already said, a strong point and necessary requirement of a functional rehabilitation software is precisely variability, which can only be achieved by

basing the entire system on a stimuli repository as wide as possible to avoid repetitiveness and consequent drops in patients' compliance. Besides, it is of vital importance to be able to handle several specific domains. For example, in the current version of the ontology, among the available "container-concepts" are:

- animals
- plants
- clothes
- jobs
- places
- foods

It can be noticed how hard it is to combine all these families in one ontology, all the while respecting the requirement of single inheritance (a concept is only allowed to have one parent / be subsumed by only one class).

### 3.1.1. Classification criteria and relations

To solve the problem, the criterion adopted to create the first structure was linguistic: the level of the taxonomy immediately following the default "THING" root-class, is made up of concepts representing basic elements of the Italian language: "*verb*", "*adjective*", "*noun*", "*article*" and "*letter*" (plus the class "*number*"). This way, despite the heterogeneity of the families to be included, it is possible to group all of them under a common container-namely, under the class "noun". This particular- and apparently not very agile- classification is actually useful when it comes to grammar/syntax based tasks, very common in rehabilitation of speech impairments.

After the first level, created mainly to aggregate all the families in the same ontology, other classifications must be introduced to further articulate the tree structure. And as it can be deduced just by looking at the list above, it is not possible to apply the same criterion to all of the items.

Some of those classes can be structured based on scientific methods: animals and plants are in fact classified according to a Linnean-like taxonomy (simplified systematic-like classification for animals, strict Cronquist classification for the plants [40]- see fig. 24), with levels representing kingdoms, *phyla*, classes, orders, *geni*, species etc. This way it is possible to prevent problems (though for some of the plants' classification levels some uncertainties and conflicts arise, because of the coexistence of several different- and officially acknowledged-classifications) related to user-defined classification criteria.

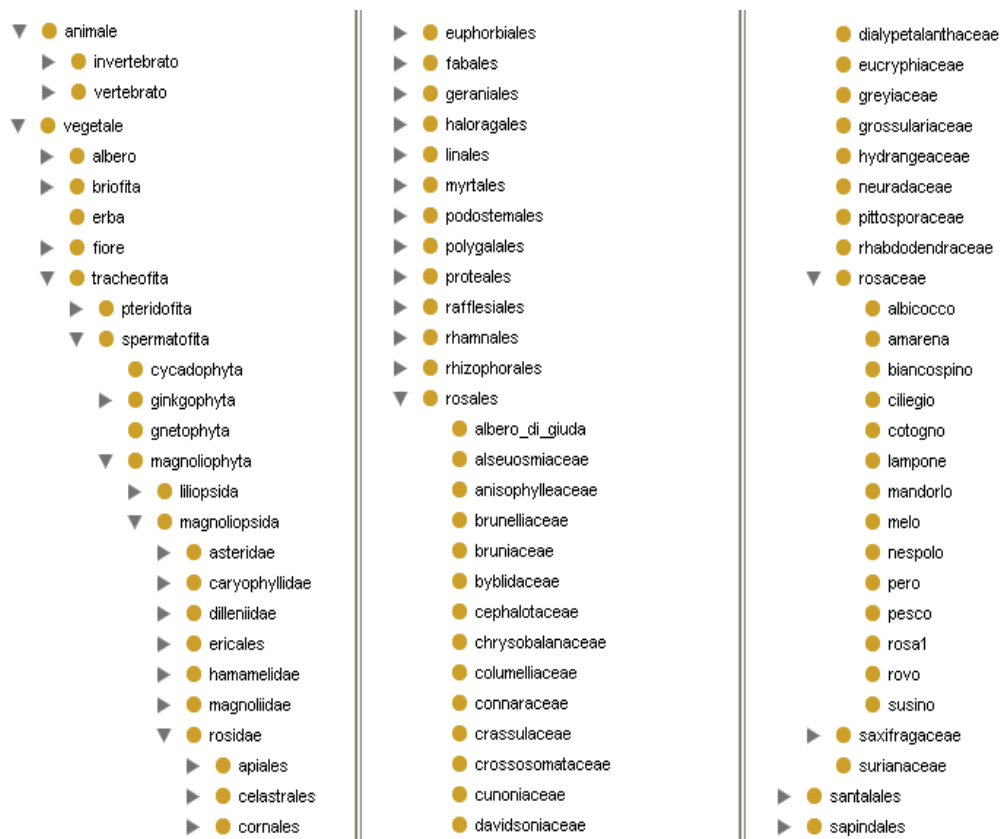


Figure 24: Excerpt of the actual ontology. Structure of the Cronquist classification of plants

Doubts might arise about the utility of such a complex and scientifically precise classification. The goal is the patient-tailoring of the rehabilitation, a topic that will be discussed in detail in chapter 3.3, dedicated to personalization.

Another particular classification criterion is the one involving jobs; two separate branches have been included into the ontology tree: the first based on the ISTAT classification of professions (representing the leafs with the complete denomination of each job- e.g. "*artisan, mechanic repairer or maintenance technician of automobiles*"), the second containing the simplified names of the jobs (the simplified equivalent to the previous example would be "*mechanic*"). The entries belonging to the two separate branches are then interconnected with a "*simplified\_profession*" relation (fig. 25).

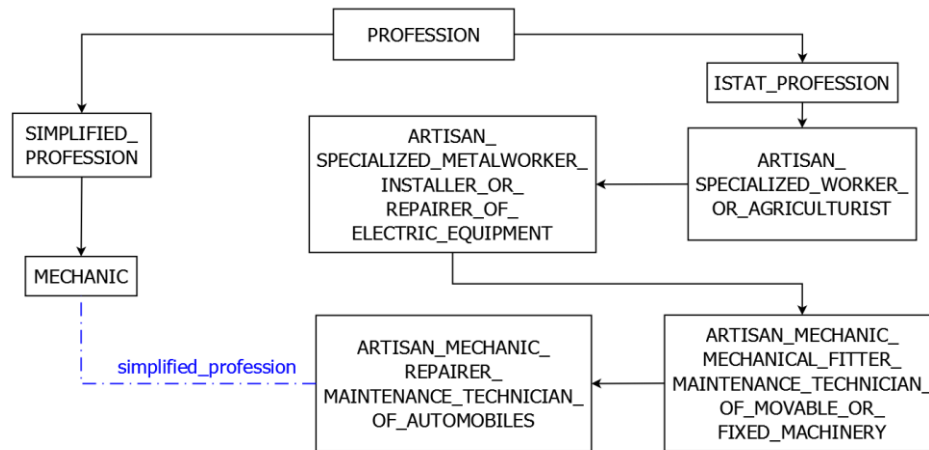


Figure 25: Translated excerpt of the ontology. Example of correspondence between professions vs. simplified professions

For other families, though, customized categorizing can't be avoided: the classification is purely "target-oriented", created following the strategy that best fulfills the requirements and needs that emerge during the creation of the exercises. For example, the taxonomy of foods has been structured taking into consideration the course type of the single elements, which is an important aspect for some of the exercises. The classes "*appetizer*", "*main course*", "*side dish*" and "*dessert*" have been created, and every leaf concept of the "*food*" tree is linked to one or more of these classes through the relation "*is\_served\_as*". Also, the class "*recipes*" has been introduced, subsuming entities representing actual, very simple culinary preparations (e. g., "*spaghetti with tomato sauce*", "*fish and chips*" etc). Of course, all the ingredients' classes are related to the corresponding recipes through the mereology relation "*is\_an\_ingredient\_of*".

An interesting element emerged during the creation of the ontology: in some cases, classes can be found that belong to different branches of the schema (and thus should be considered as separate entities), but actually represent different aspects of the same concept (this would introduce the multiple inheritance- something we want to avoid in our ontology). A recurring example in our ontology involves animals: for example, the class "*tuna*" can be found under the "*vertebrate*"> "*fish*"> "*saltwater\_fish*" category, but also as a leaf in our "*food*" sub-tree. This could cause confusion and incorrect interpretations of tasks during the rehabilitation sessions.

For this reason, keeping in mind that we chose to avoid introducing instances in our schema and only use classes instead, as well as not to allow multiple inheritance (and that, for this reason, the same "*tuna*" cannot be a subclass of both the "*animal*" and the "*food*" classes), it has been necessary to introduce the user-defined relationship "*same\_concept\_as*", to connect classes which show this characteristic. An example that explains this mechanism in detail is shown in fig. 26.

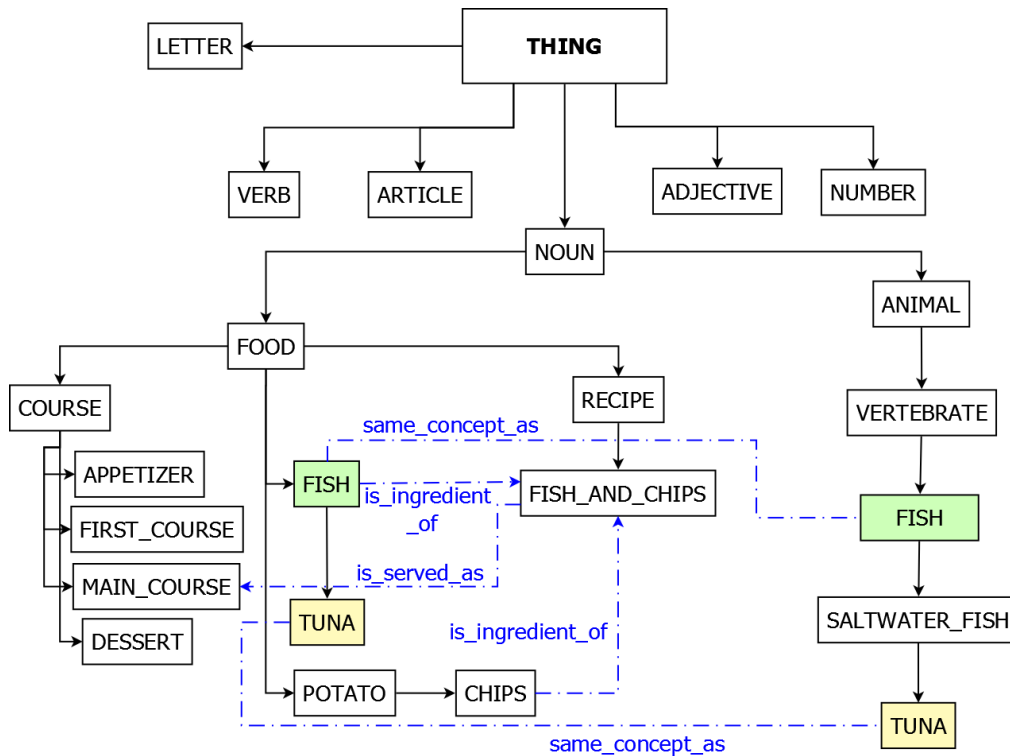


Figure 26: Translated excerpt of the ontology. Schema illustrating the case described above

## 3.2. Ontology representation

The software tools chosen for the task of representing the ontology, as well as the structure and characteristics of the resulting schema are described next.

### 3.2.1. Protégé

Protégé is an open source platform developed by Stanford University; it is a Java-based editor for the creation of ontology, able to export schemas generated in different formats, like RDF(S), XML Schema and OWL.

Its main user interface appears as shown in fig. 27:

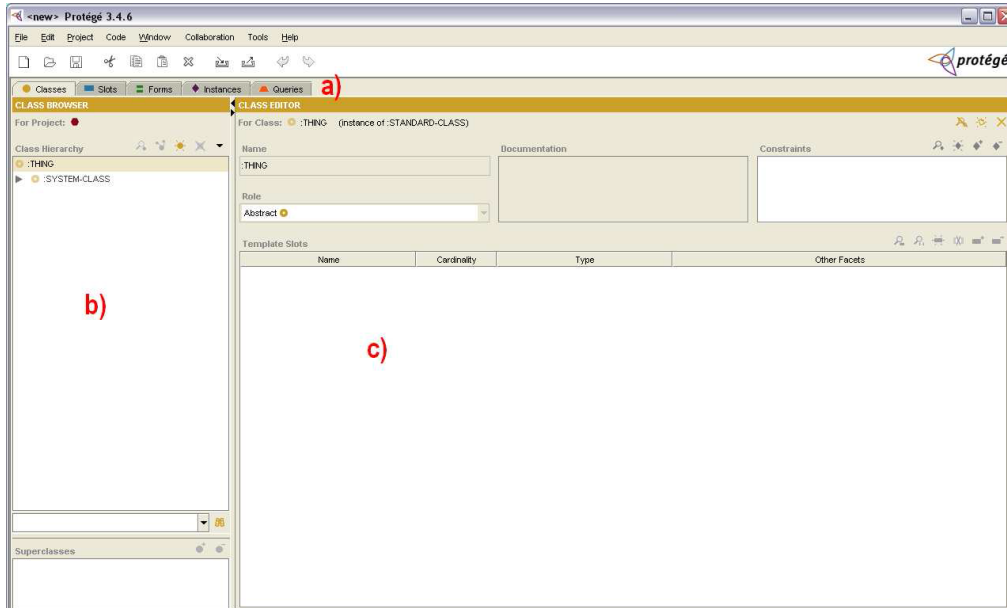


Figure 27: Main GUI of the Protégé software

- the tabs (a) that give access to different parts of the interface;
- the *class browser* (b), where the hierarchic structure of the ontology is visible;
- the *class editor* (c), that allows the assignment of values to parameters related to the selected class.

Within the class browser (shown in figure 28), subsumption relations between concepts are managed ("*is\_a*" parent-child relations).

This component is made up of two elements:

- *class hierarchy* (fig. 28, top), that allows modifications of the taxonomy through drag and drop of elements on its tree representation;
- *superclasses* (fig. 28, bottom), where the parent entities of the selected element are shown. It is possible to create new subsumptions just by manually associating new parent classes to the selected elements.

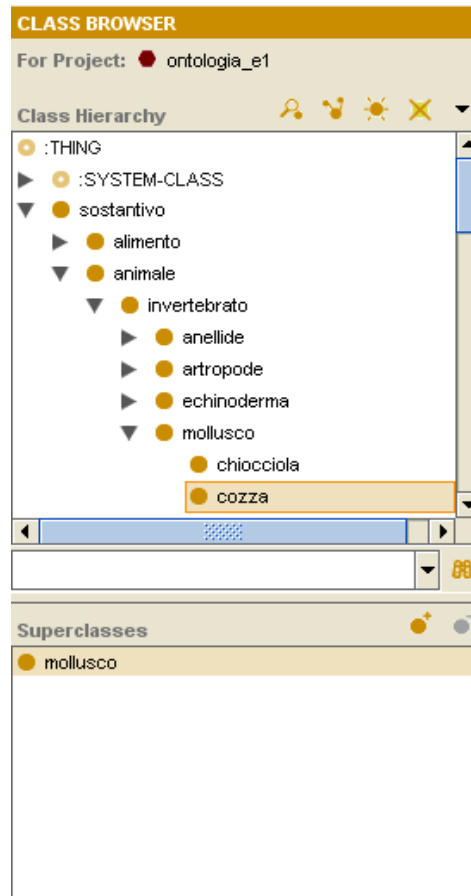


Figure 28: Components of the *Class browser*: *Class hierarchy* (top) and *Superclasses* (bottom)

Creation of new classes to be added to the ontology can be performed just by clicking a button in the class browser interface. As it can be seen in fig. 28, all the root classes- that is, the ones who lack an explicitly defined parent- are set as children of the *:THING* class (created by default as the root of the whole ontology).

For each class introduced, the Class editor GUI (shown in figure 29) makes it possible to define some parameters:

- name of the class;
- role, that is the concrete or abstract nature of the concept expressed by the class;
- constraints: the restrictions related to the class
- slots, that can represent properties of the class or relations between concepts.

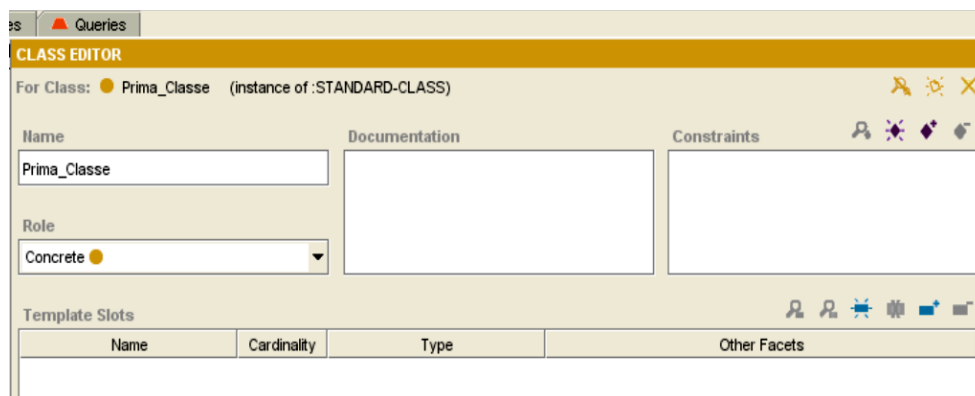


Figure 29: Protégé's "Class editor" component

Similarly to what happens in classes, also slots allow the definition of parameters (fig. 30):

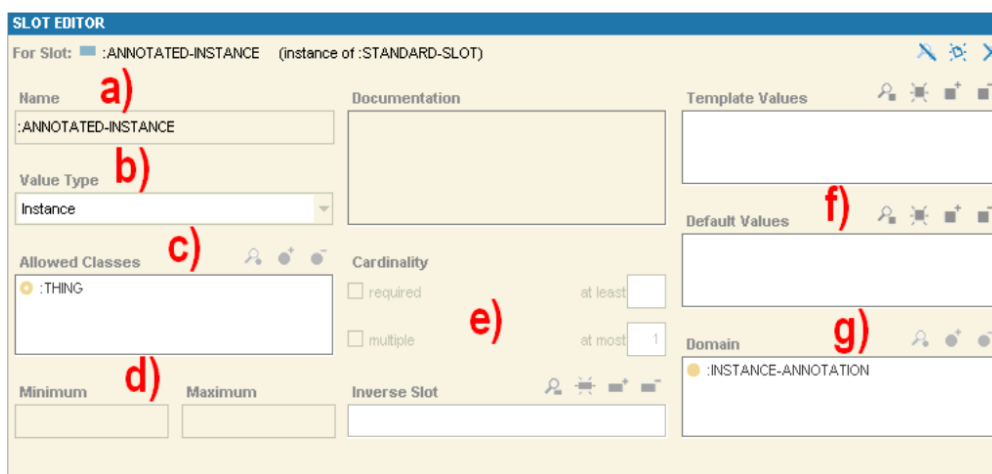


Figure 30: Example of an empty Slot editor, used to create properties and relations

- name (a);
- type (b), where it can be defined whether the slot represents a property or a relation;
- allowed classes (c): if the slot is a relation, this field specifies the linked class;
- min/max allowed values (d) (for numeric slots);
- cardinality (e), to set the maximum allowed multiplicity of the values;
- template values and, if necessary, a default value (f) to assign to the property;
- domain (g): the class to which the slot belongs.



It is possible to assign as many slots as needed to a class, in order to define its properties and the relations that connect it to other concepts of the ontology.

### 3.2.2. The XML structure

When an ontology is implemented using the tools described above- that is, creating a tree-like structure of classes characterized by properties and relations that link these concepts to each other- Protégé makes it possible to store all the information in an XML file. Below, the structure of our ontology's XML is analyzed.

After the first line, where the location of the protege.xsd file- that is, the XML Schema in which the structure of the ontology xml is detailed- is specified, the class definition starts.

The first element enclosed into the `<class>` tags is the default `:THING` class, which, as already said, represents the root of the ontology; after that, other 4 default classes are defined (`:SYSTEM-CLASS`, `:STANDARD-CLASS`, `:STANDARD-SLOT` and `:PAL-CONSTRAINT`).

Then the user-defined content can be found: analyzing the content of any class, it is possible to understand not only its exact position within the ontology (one of the first elements of a concept is `<superclass>`, where its parent is specified) but also what attributes have been associated to the classes of our ontology (fig. 31).

```

- <class>
  <name>lettera</name>
  <type>:STANDARD-CLASS</type>
- <own_slot_value>
  <slot_reference>ROLE</slot_reference>
  <value value_type="string">Concrete</value>
</own_slot_value>
<superclass>:THING</superclass>
<template_slot>descrizione</template_slot>
<template_slot>inclusione</template_slot>
<template_slot>art_p</template_slot>
<template_slot>art_s</template_slot>
<template_slot>difficoltà</template_slot>
<template_slot>freq_uso</template_slot>
<template_slot>nome_p</template_slot>
<template_slot>nome_s</template_slot>
<template_slot>sillabe_p</template_slot>
<template_slot>sillabe_s</template_slot>
<template_slot>genere_p</template_slot>
<template_slot>genere_s</template_slot>
- <template_facet_value>

```

Figure 31: Excerpt of the class `lettera`, where the property slots are listed. `Lettera` is a root-class of our ontology: this can be deduced seeing that its `<superclass>` is `:THING`.

- *descrizione* is a text field used to detail the nature of the concept (especially when namesake classes representing different concepts);
- *inclusione* is a flag that indicates whether the class must be included during the dynamic generation of the exercises. By default the flag is set to True, but if the concept is deemed unsatisfactory by the therapist, the value can be quickly and easily switched to False;
- *nome\_p/s* are two fields where the text stimuli expressing the plural and singular version of the concept are stored. For example, the class named "uomo" (*man*) has "uomo" and "uomini" (*men*) as values for *nome\_p* and *nome\_s*.
- *art\_p/s* is the definite article associated to *nome\_p/s* (for the class "uomo", they would be "l " and "gli")
- *difficoltà* is a numeric attribute that representing the complexity of the concept expressed by the class (integer value between 0, "easy", and 3, "very difficult")
- *freq\_uso* expresses the word frequency of the text stimuli (*nome\_p/s*). This information is obtained from GraDIT (**GR**Ande **D**izionario **IT**aliano dell'uso), which classifies every term under one of the following categories:
  - high frequency/use
  - high availability/familiarity
  - common
  - technical/specialized use
  - literary use
  - non-adapted foreign terms
  - regional use
  - dialectal
  - low use
  - obsolete
- *sillabe\_p/s* is the number of syllables of the text stimuli (respectively 2 and 3 for the example "uomo"/"uomini")
- *genere\_p/s* specifies the grammatical gender of terms. It is specified for both the plural and the singular term to include exceptions that change gender together with the grammatical number (e.g. "uovo"/"uova").

It can be noticed that a class includes attributes for the singular and plural declination of the concept, but not for the masculine and feminine. This choice is based on a simple observation: while the singular and plural version of a concept only show a difference from the point of view of the grammar, but maintains the concept unchanged (e.g. *uomini* can be defined as "more specimens of *uomo*"), it is not infrequent that the feminine and masculine versions of a general concept express entirely different entities (e.g. "*madre*", *mother*, is not "the feminine version of *padre*, *father*", neither from the lexical nor the conceptual point of view). For this reason,

different grammatical genders are translated in separate classes whenever possible.

Below, fig. 32 shows the class "leone" (*lion*), where all the attributes are compiled and some relations are featured:

```

- <class>
  <name>leone</name>
  <type>:STANDARD-CLASS</type>
- <own_slot_value>
  <slot_reference>:ROLE</slot_reference>
  <value value_type="string">Concrete</value>
</own_slot_value>
  <superclass>placentato</superclass>
  <template_slot>è di colore</template_slot>
- <template_facet_value>
  <slot_reference>abita</slot_reference>
  <facet_reference>:VALUES</facet_reference>
  <value value_type="class">tana</value>
</template_facet_value>
- <template_facet_value>
  <slot_reference>abita</slot_reference>
  <facet_reference>:VALUE-TYPE</facet_reference>
  <value value_type="string">Class</value>
  <value value_type="class">abitazione_per_animali</value>
</template_facet_value>
- <template_facet_value>
  <slot_reference>si_ciba_di</slot_reference>
  <facet_reference>:VALUE-TYPE</facet_reference>
  <value value_type="string">Class</value>
  <value value_type="class">gazzella</value>
</template_facet_value>
- <template_facet_value>
  <slot_reference>si trova in</slot_reference>
  <facet_reference>:VALUE-TYPE</facet_reference>
  <value value_type="string">Class</value>
  <value value_type="class">savana</value>
</template_facet_value>
- <template_facet_value>
  <slot_reference>art_p</slot_reference>
  <facet_reference>:VALUES</facet_reference>
  <value value_type="string">i</value>
</template_facet_value>

```

```

- <template_facet_value>
  <slot_reference>art_s</slot_reference>
  <facet_reference>:VALUES</facet_reference>
  <value value_type="string">il</value>
</template_facet_value>
- <template_facet_value>
  <slot_reference>difficoltà</slot_reference>
  <facet_reference>:VALUES</facet_reference>
  <value value_type="integer">1</value>
</template_facet_value>
- <template_facet_value>
  <slot_reference>nome_p</slot_reference>
  <facet_reference>:VALUES</facet_reference>
  <value value_type="string">leoni</value>
</template_facet_value>
- <template_facet_value>
  <slot_reference>nome_s</slot_reference>
  <facet_reference>:VALUES</facet_reference>
  <value value_type="string">leone</value>
</template_facet_value>
- <template_facet_value>
  <slot_reference>sillabe_p</slot_reference>
  <facet_reference>:VALUES</facet_reference>
  <value value_type="integer">3</value>
</template_facet_value>
- <template_facet_value>
  <slot_reference>sillabe_s</slot_reference>
  <facet_reference>:VALUES</facet_reference>
  <value value_type="integer">3</value>
</template_facet_value>
- <template_facet_value>
  <slot_reference>è_di_colore</slot_reference>
  <facet_reference>:VALUE-TYPE</facet_reference>
  <value value_type="string">Class</value>
  <value value_type="class">giallo</value>
</template_facet_value>
</class>

```

Figure 32: Excerpt from the actual ontology. The class "leone", complete with all of its properties and relations

From the figure above, it is possible to understand that a "leone" (a lion):

- lives in a den ("abita" -> "tana")
- eats gazelles ("si\_ciba\_di" -> "gazzella")
- can be found in the savannah ("si\_trova\_in" -> "savana")

- its main color is yellow ("è\_di\_colore" -> "giallo")

In the XML file, relations, just like the properties, are elements: the difference lies in the content of the "value\_type" attribute, which, for a relation, is "class". The last section of the file is dedicated to the definition of the slots' structure: all the relations and properties are explained; cardinalities, as well as default and template values (and their types) are detailed. For example, the "art\_p" slot contains the explicit definition of three elements, "i", "gli", "le", which are the template values of the property)

```

</class>
- <slot>
  <name>abita</name>
  <type>:STANDARD-SLOT</type>
  - <own_slot_value>
    <slot_reference>:SLOT-VALUE-TYPE</slot_reference>
    <value value_type="string">Class</value>
    <value value_type="class">abitazione</value>
  </own_slot_value>
</slot>
- <slot>
  <name>art_p</name>
  <type>:STANDARD-SLOT</type>
  - <own_slot_value>
    <slot_reference>:SLOT-MAXIMUM-CARDINALITY</slot_reference>
    <value value_type="integer">1</value>
  </own_slot_value>
  - <own_slot_value>
    <slot_reference>:SLOT-VALUE-TYPE</slot_reference>
    <value value_type="string">Symbol</value>
    <value value_type="string">i</value>
    <value value_type="string">gli</value>
    <value value_type="string">le</value>
  </own_slot_value>
</slot>
- <slot>
- <slot>
  <name>art_s</name>
  <type>:STANDARD-SLOT</type>
  - <own_slot_value>
    <slot_reference>:SLOT-MAXIMUM-CARDINALITY</slot_reference>
    <value value_type="integer">1</value>
  </own_slot_value>

```

```

- <own_slot_value>
  <slot_reference>:SLOT-VALUE-TYPE</slot_reference>
  <value value_type="string">Symbol</value>
  <value value_type="string">il</value>
  <value value_type="string">lo</value>
  <value value_type="string">la</value>
  <value value_type="string">l'</value>
</own_slot_value>
</slot>
- <slot>
  <name>compie</name>
  <type>:STANDARD-SLOT</type>
  - <own_slot_value>
    <slot_reference>:SLOT-MAXIMUM-CARDINALITY</slot_reference>
    <value value_type="integer">1</value>
  </own_slot_value>
  - <own_slot_value>
    <slot_reference>:SLOT-VALUE-TYPE</slot_reference>
    <value value_type="string">Class</value>
    <value value_type="class">azione</value>
  </own_slot_value>
</slot>

```

Figure 33: Excerpt of the actual ontology. Definition of properties and relations

### 3.2.3. The XML-SQL translator

After the ontology has been populated, a complete knowledge base is available for use in the system. The problem is, while the XML-based format is the best and easiest in the population phase, the same doesn't apply to data retrieval from said structure. Thus, the aim is to transfer all the information from the ontology into a relational database that can easily be integrated into the system (stimuli\_db).

For this reason a software translator has been created, that scans the XML file gathering all the information available for each element and then, through appropriate SQL queries, populates the database accordingly (fig. 34).

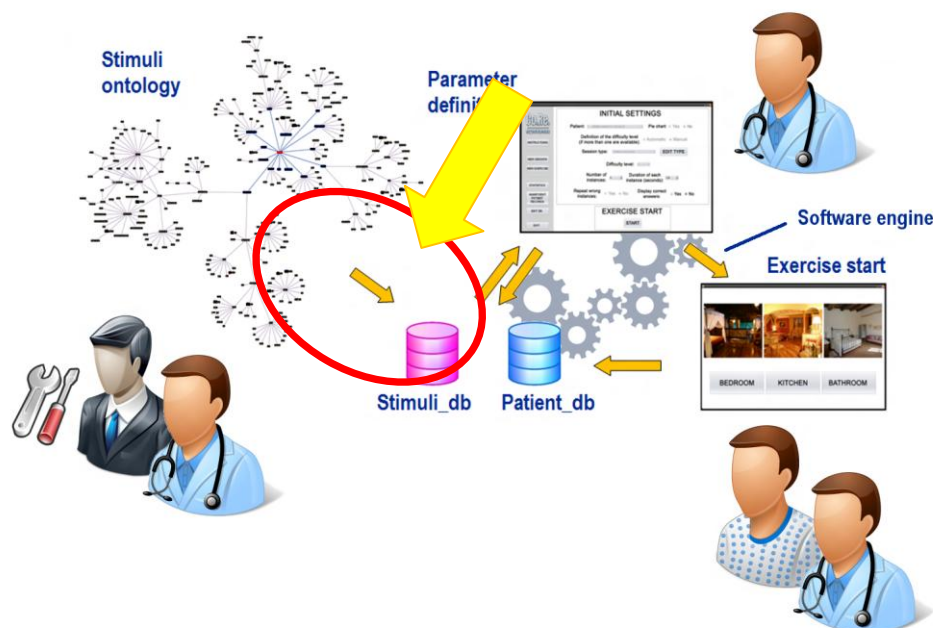


Figure 34: Location of the XML-SQL translation tool in the system structure

This java software is made up of 5 classes:

- Main
- TuplaOntologica (TO)
- TuplaRelazione (TR)
- MatriceOntologica (MO)
- MatriceRelazione (MR)

The TO and TR classes represent the concepts and relations of the ontology, which are transferred as records into two separate tables of the new database. Of course, considered that a concept of the ontology contains more than a stimulus, the correspondence between elements of the XML file and records of the final database is "one-to-many" (that is, an element of the ontology translates into multiple entries of the database).

The properties of the TO/TR classes are the attributes to be saved in the tables, while the methods allow the instantiation of a tuple with attributes that are either empty or set as parameters of the constructor method.

Classes MO and MR handle ArrayLists of TO and TR objects. These classes have constructors and methods that allow the insertion of new TO/TR objects in the matrix and search of specific tuple (given that key attributes are known and used as the search parameters). Methods of database population are also present, that scan the matrices and create queries to insert data into the corresponding tables.

Finally, the Main class starts the program and organizes data before the translation: it reads the XML file, assigns a mnemonic code (that reflects

the hierarchical position of the element) and extracts attributes and relations.

The flowchart of the tool is presented in figure 35: after instantiating empty MO and MR objects, the default first entry is created (:THING); after this, the XML file is loaded to search other concepts and relations. Once the end of the XML is reached and all the concepts and relations have been inserted into MO and MR, the records are transcribed into two raw tables. A further phase of refining is necessary to get to the final structure of the database.

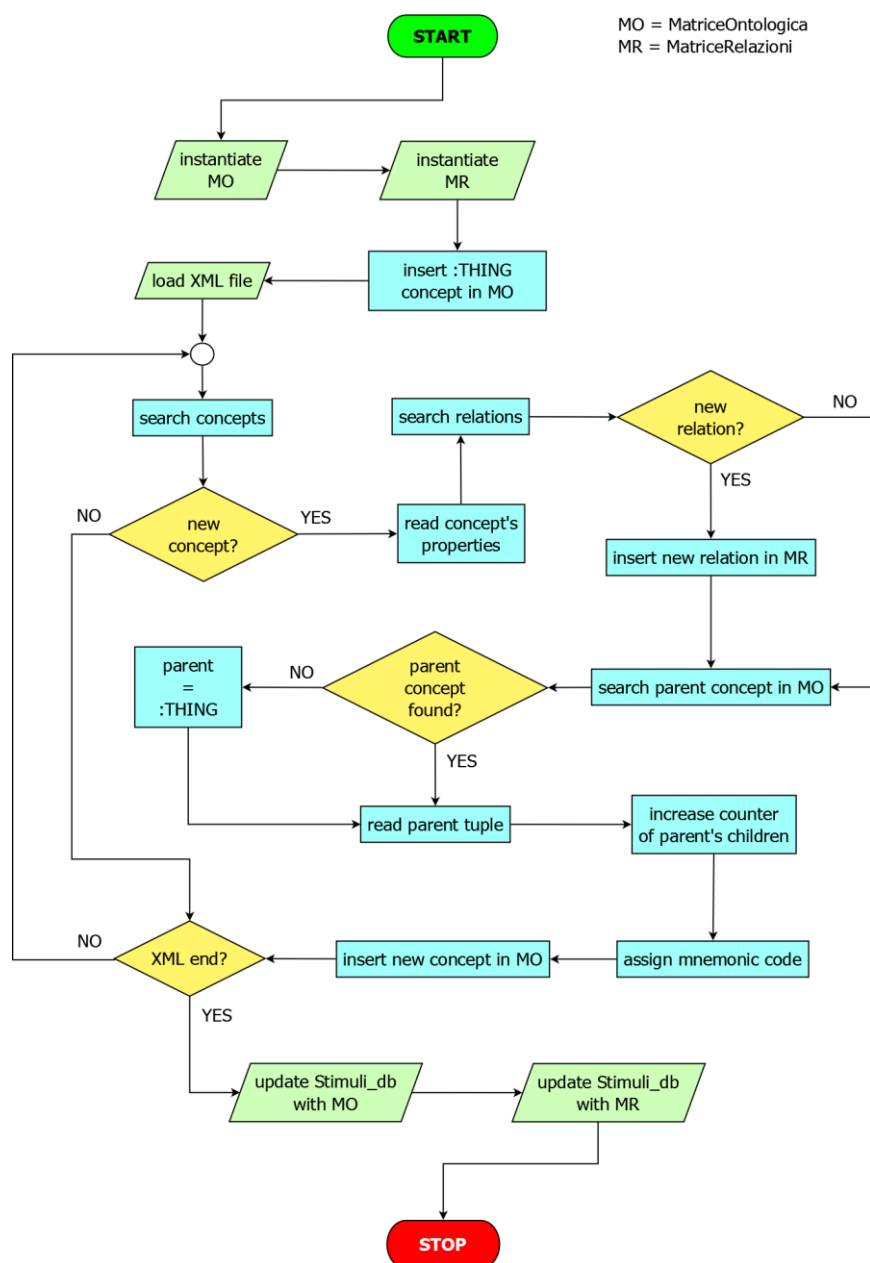


Figure 35: Flowchart of the XML-SQL translator. The update of the database is the last step, performed after the ontology has been completely transferred to MO and MR



### 3.3. Personalization

An important characteristic of our system is customizability; the attempt to achieve this goal is one of the reasons behind the choices we made during the project phase of the ontology.

- The "difficulty" property, attached to every concept is the most obvious: by classifying the elements based on their intrinsic complexity it is possible for the software engine to automatically calibrate the selection of stimuli to be displayed during the execution of the tasks.
- The same reasoning applies to the "word frequency": under the assumption that for a term, belonging to the "easier" groups ("high frequency/use", "high availability/familiarity" and "common") implies a lower difficulty, it is possible to also use this parameter to personalize the sessions. But a problem arises when terms are classified under other categories, because no judgment can really be expressed about those concepts' difficulty. In fact, considering the type of patients at which our system is targeted, not necessarily a "dialectal", "regional" or "obsolete" term is to be considered difficult. For this it is not possible to rely solely on the word frequency classification: the matter is more complex, and must be dealt with keeping into consideration the patient's background, which leads to the third item of this list.
- The higher level of personalization consists in a more complex approach and requires merging data from the ontology with patients' personal information stored into the database. As hinted in the previous paragraph, some of the word frequency categories- particularly "technical/specialized use"- can't be associated to a specific level of difficulty by default. This type of classification depends largely on the patient's scholarship and hobbies. For example, the word "vertebrate" (defining animals' characteristic of possessing a backbone) doesn't fall into the "high frequency" category, and thus should be classified as difficult, since it would not be easily recognizable by the average patient. But for a subject with medium-high level of scholarship, or even dealing with animals- because of his job or a hobby- the term would be very familiar, maybe even a basic notion.

To solve this impasse it is necessary to introduce a new method of classification based on:

- 1) "tagging" these terms in the ontology according to the domain(s) they belong to;
- 2) organizing the patient's personal profile, creating a well-structured questionnaire based on controlled vocabulary. The answers available for each entry are also tagged, so that the choice of an answer during

the compiling of the questionnaire sets a rule for the selection of stimuli during the generation of the exercises.

For example, a patient answers the therapist's request saying that he works as an accountant: the "occupation" field of his profile (a drop-down menu whose options are derived from the "*simplified\_professions*" records of the database) will be compiled accordingly.

Every item of the menu has its tags associated; for example, the term "*accountant*" is tagged as <economy> and <law>. A scan of the database reveals that other entries are similarly tagged. This way, when the exercises are generated, stimuli representing concepts that would normally be considered extremely difficult or specialized and thus excluded from the selection, are included.

This type of personalization is currently under development and will be available in the next implementation of CoRe (as mentioned in chapter 7).

---

# Chapter 4

---

## The CoRe system

### 4.1. First version exploiting commercial tools

#### 4.1.1. E-Prime

The E-Prime® [41][42] application suite, used within the first version of CoRe, is one of the most powerful and flexible tools available on the market.

Its strong point, and also the characteristic that made it our software of choice, resides in not supplying a fixed set of exercises: E-Prime® makes it possible to create many types of exercises (called "experiments") in a simple and immediate way.

It features several tools:

- *E-studio*, a development environment which makes it possible to define experiments through the combination of predefined objects, the so-called "E-Objects", and scripting in *E-Basic*, the programming language used within E-Prime;
- *E-Run*, the application that creates the executable instance of the experiment and allows the synchronization of events and the data collection;
- *E-Merge*, combining data coming from different rehabilitation sessions;
- *E-Data Aid*, able to filter, edit, analyze and export performance data;
- *E-Recovery*, that retrieves data in case of unexpected interruption of an experiment or existence of corrupted files.

Within CoRe, in use at the Neuropsychology Center / Alzheimer Evaluation Unit of the IRCCS Mondino Foundation, only E-Studio, E-Run and the language E-Basic are used, because patients' performance data are stored into databases, handled by CoRe's software engine and visualized into a Visual Basic Graphic User Interface (GUI).

#### 4.1.1.1. E-Studio

As already mentioned, E-Studio is the environment for the creation of the experiments. Its interface features five principal tabs:

- *Toolbox (a)*, the container of the E-Objects, the blocks necessary for the creation of the exercises.
- *Structure (b)*, where the hierarchical structure of the components of the experiment is shown.
- *Properties (c)*, where the parameters associated to the selected E-Object can be edited;
- *Output (d)*, that shows the result of the generation, compiling and debug;
- *Workspace (e)*, where the Objects' representations are collected;
- *Script (f)*, which contains all the E-Studio-generated code at the start of E-Run.

In order to set the global properties of an experiment, it is necessary to select the item called "Options" from the "Tools" drop-down menu.

The resulting pop-up window features several tabs:

- *General*, where the experiment path and the execution parameters (for the compilation and generation of the scripts) can be set.
- *Structure, Properties, Attributes* and *Output*, allowing the definition of parameters regarding the structure of the window;
- *Toolbox*, useful to specify parameters related to the toolbox. By default all of the available E-Objects are shown;
- *Script*, to set the syntax colors for the code.

The creation of an experiment with E-Studio implies the manual composition of every single screen (the same which are visualized by the patient during the execution).

The procedure is easy and fast, based on the drag-and-drop of the elements: "basic" E-Objects (among the available types, *TextDisplay*, *ImageDisplay* and *SoundOut* have been used within CoRe) are placed inside specific objects acting as containers, the *Slides*, which in turn are placed into a *Procedural Timeline* (an E-Object itself), which defines the chronological order of the events.

Each one of the available E-Objects is characterized by a set of specific properties, necessary for its functioning. The most important E-Objects will be explained below:

#### 4.1.1.1.1. Experiment

Automatically generated whenever a new exercise is created, it represents the root of the hierarchical structure of the experiment. Double clicking this object triggers the visualization of a window through which it is possible to set all the global properties of the experiment, among which:

-*Startup info*, used at the beginning of the exercise to define information about the patient (by default his personal id and the session number). Actually, this property is not used within CoRe: the acquisition of these (and other) parameters is automated and handled via read/write operations to/from .txt files executed by the so called *Inline Objects*.

-*Data file*, allowing the user to set parameters regarding the creation of reports. It warns the user when a report is about to be overwritten and notifies the conversion of a text file in an E-Prime Data File (.EDAT).

-*Devices*, that enables and initializes I/O peripherals used in the experiment. Within CoRe the standard I/O hardware is used (and therefore must be enabled) : keyboard, mouse, screen and audio output.

-*Packages*, used to install or remove an E-Prime Package File, allowing the user to import previously created scripts and subroutines.

#### 4.1.1.1.2. Procedure

It is used to define a chronological sequence of events in the execution of the exercise. It features a *Procedural Timeline* (fig. 36), a visual component representing a line on which the objects are dropped after being dragged from the *Toolbox*.



Figure 36: Example of a Procedure. Each icon represents an Object; the function of all the elements is explained in this chapter

It is possible to have nested procedures within an experiment. To be iterated, every procedure must be connected to a *List* type E-Object.

#### 4.1.1.1.3. List

This object contains data which are used within the Procedure to which it is associated and all its sub-procedures. Besides the default attributes (called *Weight*, *Procedure* and *Nested*), needed for the experiment to function correctly, other user-defined parameters can be introduced.

Values can be inserted manually or, as it happens in CoRe, loaded from a tab-delimited text file; the number and type of the attributes vary with the exercise type, but three parameters recur in all the files:

- *Stimulus*, containing the stimuli presented in an instance;
- *StateName*, indicating the case (configuration of objects into a *Slide*) to visualize;
- *CorrectAnswers*.

Its properties are:

- *Common*, to edit the object's name;
- *General*, to set the data loading method: "embedded" (from a table), "file" (from a tab-delimited text file) or "script";
- *Selection*, that determines the loading order of the data;
- *Reset/exit*, to set the duration and number of iterations for the selected list and to determine the exit condition.
- *View*, to set the visualization mode of the List attributes;
- *Logging*, to select specific attributes (response time, total duration of the experiment, number of correct/wrong answers) to be inserted into the report.

Many of the properties explained above recur in other object types; therefore they will not be mentioned again, unless significant for the functioning of the object itself.

#### **4.1.1.1.4. TextDisplay**

Used to create screens with exclusively textual content: within CoRe exercises, it is used to create the Welcome page, the description/instructions page and the goodbye message screen.

Among its properties:

- *Duration/Input* allows the user to set the maximum amount of time (milliseconds) allowed for visualization of this screen, as well as the synchronization method ("on event", "cumulative" or "custom");
- *Sync* controls the synchronization of the start and end of this object with specific events, like the refresh of the screen.

#### **4.1.1.1.5. ImageDisplay**

This object is used to show a screen containing nothing but an image (.bmp and .jpg are the compatible formats). From the *General* property menu it is possible to specify the path of the image to visualize.

#### 4.1.1.1.6. SoundOut

This object is used to play digital audio files within an experiment. The property General is useful to set the parameters regarding the sound file: not only its name, but also the maximum size of the buffer and the playback offset.

#### 4.1.1.1.7. Slide

This object is used to present multiple stimuli or combinations of texts, images and sounds. It features one or more *SlideState Objects*, each of which represents a configuration of the screen (fig. 37)

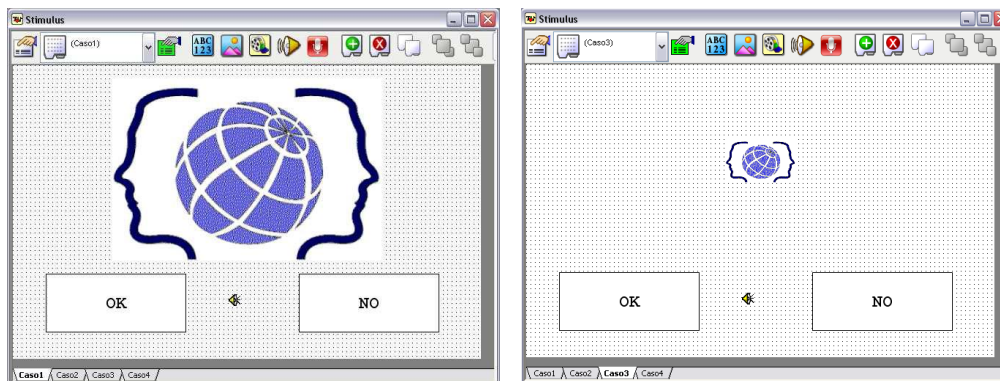


Figure 37: Example of a Slide with more SlideState Objects: the size of the image changes in different states

The *SlideState Object* to show when the experiment is executed can be specified in the *General* property menu.

The types of the *Sub\_Objects* mirror the ones already seen for the *Objects*; also, many *Sub\_Objects* can coexist in the same *SlideState*. For this reason it is possible to create composite screens in which textual, visual and audio *Sub\_Objects* are featured, as shown in fig. 38.

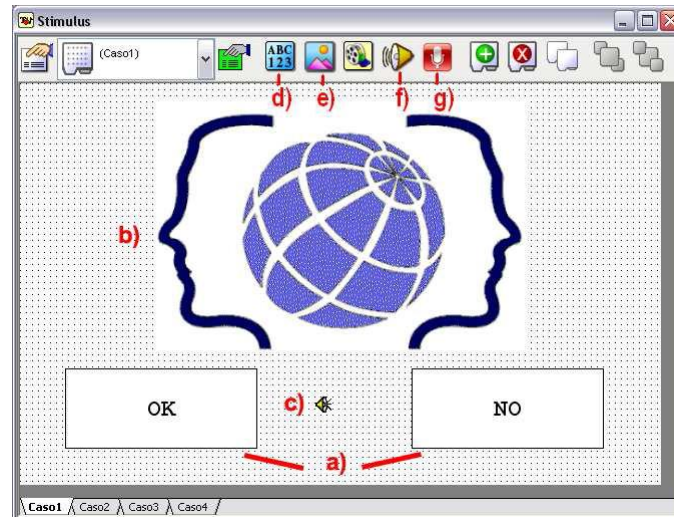


Figure 38: Different types of Sub\_Objects are available for use within a SlideState: textual (a, d), visual (b, e) and audio (c, f, g).

#### 4.1.1.1.8. FeedbackDisplay

This object is used to show a feedback at the end of every exercise. It inherits the same objects of the *Slide* it is linked to, including the *Sub\_Objects* with their properties. An example of *FeedbackDisplay* is shown in fig. 39.



Figure 39: Example of FeedbackDisplay

From the *General* property menu it is possible to define which *Slide* is linked to the *FeedbackDisplay* (that is, the *Slide* during whose execution the acquisition of performance data -like response time and number of clicks- must be performed), as well as the type of data to collect.



#### 4.1.1.1.9. Inline

It is used to insert E-Basic script in the experiment. Coding within an experiment is necessary when the requirements for an experiment are beyond the default functions of the Objects.

The main advantage of E-Prime is its flexibility, that is the possibility to generate experiments dynamically.

For every object, all the properties that can be set through free-text input can be assigned a variable (by typing the [variable\_name] into the desired text field). This way the value of the property is derived from the field of the same name that can be found in the tab-delimited .txt file set as a source for the *Procedure* (fig. 40).

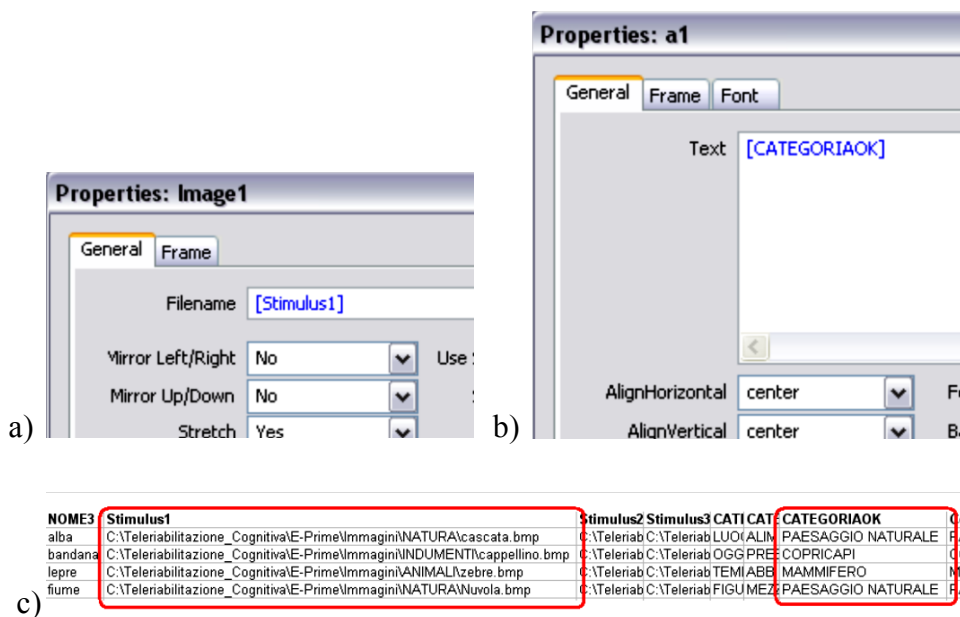


Figure 40: Example of dynamic content for an ImageDisplay Sub\_Object (a) and a TextDisplay Sub\_object (b). The values of the parameters are derived from the fields of the "DATI.txt" file

#### 4.1.1.2. E-Run

Every time an experiment is generated, the .es2 file created by the developer using the E-Studio IDE is automatically converted in a .ebs file, needed for the execution. This file is loaded by the E-Run engine right before the start of the exercise, which can then be executed by the patient.

### 4.1.2. System architecture

CoRe features four main components:

- a Visual Basic interface allowing the user (that is, the therapist who creates the treatment plans for the patients) to set the parameters needed to start an exercise (difficulty level, number and maximum duration of the stimuli to be shown etc.);
- two MS Access databases, used to store the patients' personal information and performance parameters, as well as all the stimuli (texts, sounds and images) needed for the execution of the sessions;
- the exercises, which have been created with E-Prime;
- a software engine, implemented in Visual Basic, able to generate the customized exercises based on both the options set by the therapist and the patient's performance.

The structure of CoRe is shown in figure 41.

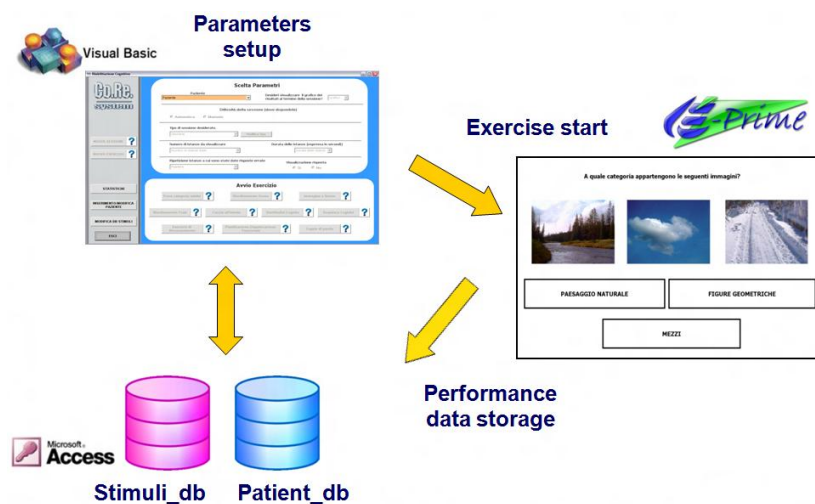


Figure 41: Structure of the first implementation of CoRe

Just after setting the parameters, the therapist can start the exercises; the databases are accessed for Create Read Update Delete (CRUD) operations and three main purposes:

- when the database must be updated, and new stimuli inserted;
- when these stimuli need to be extracted in order to generate a new session;
- when the system needs to read/write the patient's performances and personal information.

Besides the execution of exercises, accessory functions are available, that allow the therapist to interact with the stimuli database, editing it to

meet the patients' requirements. A navigation diagram, showing screenshots of the actual forms composing CoRe, is presented below.

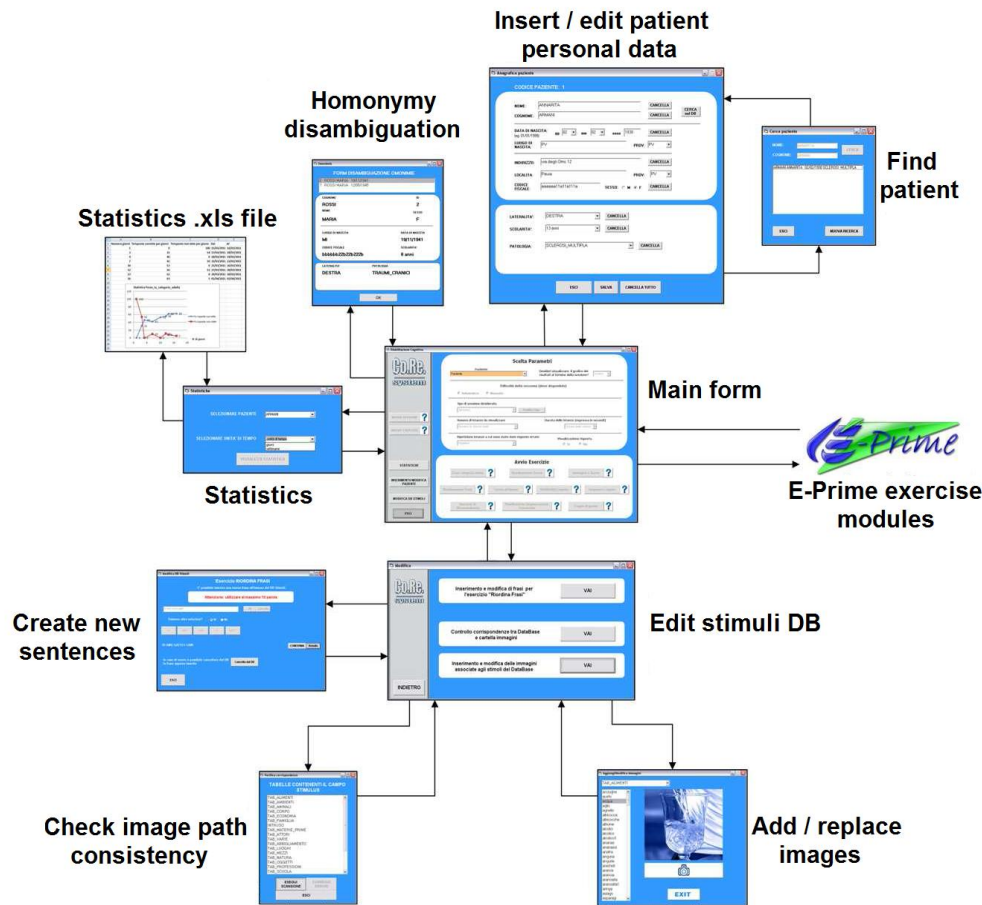


Figure 42: Navigation diagram of CoRe

### 4.1.3. The databases

As already mentioned, all the data managed by CoRe are stored in two distinct databases: the first (patient\_db) containing all the personal information regarding the patients; the second (stimuli\_db) containing the stimuli necessary to correctly generate the exercises.

#### 4.1.3.1. Patient database

The first table featured in this archive is "anagrafica", a container of personal information regarding the patients undergoing the rehabilitation. Beside standard fields like first/last name, birth date, pathology, residence address etc., each record also features a field called "id\_paziente", used to univocally identify a subject and to act as a foreign key in the database.

In the table "sessioni" (fig. 43), data concerning single executed instances are stored. It is so structured:

- *id\_paziente*;
- *data*, containing the timestamp (date and time) of the execution;
- *acc*, a 1-digit numerical field with 3 possible values: 1 for correct answer, 0 for wrong / incomplete answer, 2 for no answer (timeout);
- *totclicks*, the total number of clicks performed by the user before solving the instance (or before the timeout); this value is only present in records concerning more complex tasks requiring multiple selections to be solved, like word coupling, unscramble sentences etc. (in case of wrong choice, single-click exercises end without allowing the user to correct his selection);
- *rt*, where the response time- that is the number of milliseconds spend to solve the instance- is stored;
- *numero\_sessione*, the counter indicating to which session the instance belongs;
- *nome\_esercizio*, the name of the exercise to which the instance belongs.
- *durata\_stimoli*, where the maximum number of milliseconds allowed for the resolution is stored. Then, the instance is timed out by the system and the no answer/incomplete answer feedback is visualized;
- *livello\_difficolta*, that is the numeric indicator of the difficulty level of the exercise;
- *ripetizione*, a boolean Y/N field indicating whether the execution requires the repetition of wrong instances;
- *valore\_dati* contains a string value indicating the content of the instance, to allow a posteriori reconstruction.

id_paziente	data	acc	TOTCLICKS	rt	numero_sessione	nome_esercizio
4	02/03/2007 21.05.58	0		156	1100	Sequenze_Logiche
4	02/03/2007 22.08.45	0		1459	1101	Sequenze_Logiche
4	02/03/2007 22.09.03	1		6403	1101	Sequenze_Logiche
4	02/03/2007 22.25.01	0		143	1102	Sequenze_Logiche
4	02/03/2007 22.25.04	1		909	1102	Sequenze_Logiche
4	02/03/2007 22.32.22	0		182	1103	Esercizio_di_Riconoscimento
4	02/03/2007 22.43.12	0		541	1104	Esercizio_di_Riconoscimento
4	02/03/2007 22.44.12	0		222	1104	Esercizio_di_Riconoscimento
4	02/03/2007 22.44.24	0		178	1104	Esercizio_di_Riconoscimento
4	02/03/2007 22.55.27	0		4390	1105	Pianificazione_Organizzazione
4	02/03/2007 22.55.31	1		2995	1105	Pianificazione_Organizzazione

durata	livello_difficolta	ripetizione	VALORE_DATI
60000	2	No	SEQUENZA18
120000	3	Si	SEQUENZA16
120000	3	Si	SEQUENZA16
60000	2	No	SEQUENZA1
60000	2	No	SEQUENZA9
120000	1	No	U
60000	2	No	G
60000	2	No	E
60000	2	No	u
60000	2	No	SCIARE\$\$\$SCARPONI\$RACCHETTE\$PENNA\$PIETRA
60000	2	No	PULIRE\$\$\$STRACCIO\$SPAZZOLONE\$RUBRICA\$TELA

Figure 43: Excerpt of the table "sessioni"

A similar table, "sessioni\_info", contains general information related to the executed sessions (1:1 correspondence between sessions and records).

As shown in fig. 44, some of its fields (*id\_paziente*, *data*, *nome\_esercizio*, *ripetizione*, *difficolta*) are present in "sessioni".

id_paziente	data	numero_sessione	nome_esercizio	modalita_terminazione	stimoli_previsti	stimoli_presentati
15	28/07/2011 9.44.47	2	Caccia_intruso	-	4	4
1	26/07/2011 13.44.59	1467	Pianificazione_Organizzazione_Funzionale	-	4	4
1	26/07/2011 13.05.32	1466	Pianificazione_Organizzazione_Funzionale	-	4	4
1	26/07/2011 13.04.23	1465	Pianificazione_Organizzazione_Funzionale	q	4	1
1	26/07/2011 10.03.05	1464	Pianificazione_Organizzazione_Funzionale	-	4	4
1	26/07/2011 10.01.15	1463	Pianificazione_Organizzazione_Funzionale	-	4	4
1	26/07/2011 9.57.58	1462	Pianificazione_Organizzazione_Funzionale	-	5	5
1	29/06/2011 16.35.22	1449	Caccia_intruso	q	2	1
1	29/06/2011 16.34.18	1448	Caccia_intruso	q	2	1
1	27/06/2011 14.36.51	1447	Immagine_suono	-	3	5
1	27/06/2011 13.57.32	1446	Immagine_suono	-	4	4

ripetizione	giri_ripetizione	difficolta	tempo_tot	nr_visualizz_esempio	dati_esercizio
No	0	1	18850	1	-
No	0	4	16806	1	-
No	0	3	16587	1	-
No	0	1	2559	1	NUOTARE!!!!PINNE!!COSTUME!!SCALPELLO!!STRUMENTI
No	0	2	111817	0	-
No	0	2	67574	1	-
No	0	1	78491	1	-
No	0	1	209	1	acetol!caffel!facoltà!pastallpomodori
No	0	1	2913	1	esportatore!!maestro!!ossidollpilotal!rappresentante
Si	1	1	2771	0	-
Si	0	1	8136	2	-

Figure 44: Excerpt of the table "sessioni\_info"

The other fields are explained below.

- *modalita\_terminazione* is a char type field where the values "q", "x" and "-" are allowed. Here the information about whether the exercise has been interrupted ahead of time is stored. If so, "q" and "x" represent different interrupt motivations; otherwise, if the exercise is completed as expected, the "-" appears in this field;
- *stimoli\_previsti* and *stimoli\_presentati* store the number of instances that are expected to be visualized and the number of instances actually visualized during the execution. These values can be different in two cases:

- the exercise is interrupted ahead of time ( $\text{stimoli\_presentati} < \text{stimoli\_previsti}$ );
- the execution required the repetition of wrong instances ( $\text{stimoli\_presentati} > \text{stimoli\_previsti}$ ).
- *giri\_ripetizione* indicates how many repetition cycles have been executed;
- *tempo\_tot* is the sum of the RTs of the instances composing the session;
- *nr\_visualizz\_esempio* keeps track of how many times the exercise demo has been visualized.
- *dati\_esercizio* is filled with a string representing the content of the last visualized instance only in case of interruption ahead of time. Otherwise, the field is filled with a "-".

The table "gradi\_difficolta" describes, for every exercise type allowing more than one difficulty level, the implications of a variation in this parameter. Criteria to establish difficulty strongly depend on the structure of the exercises.

tipo_esercizio	id_esercizio	descrizione_difficoltà
Similitudini_Logiche	EX1	cambia la posizione dell'incognita
Sequenze_Logiche	EX2	cambia il numero di immagini tra cui scegliere la risposta corretta
Esercizio_di_Riconoscimento	EX3	cambia il numero di elementi da trovare nella matrice
Pianificazione_Organizzazione_Funzionale	EX4	cambiano il numero di elementi tra cui scegliere la risposta corretta e il grado di facilitazione dell'esercizio (evidenziazioni)
Riordinamento_Frasi	EX5	cambia il grado di facilitazione dell'esercizio (evidenziazioni)

Figure 45: Excerpt of the table "gradi\_difficolta"

"livello\_difficolta" contains the accuracy thresholds needed for automatic switch of the difficulty. It's composed of 4 fields:

- *livello*, allowing numeric values between 1 and 4;
- *id\_esercizio*, foreign key to "gradi\_difficolta";
- *valore*, describing the characteristic of the level;
- *soglia\_max*, containing the threshold success value above which the level changes: the system can automatically calculate the rate of success (instances with accuracy = 1) of a patient. Comparing this value with the ones in the field "soglia\_max" it is possible to define the appropriate difficulty level for the specific exercise.

livello	id_esercizio	soglia_max	valore
1	EX1	0,7	x fissa
2	EX1		x mobile
1	EX2	0,6	2 soluzioni
2	EX2		4 soluzioni
1	EX3	0,6	1 elemento
2	EX3	0,8	2 elementi
3	EX3		3 elementi
1	EX4	0,25	4 soluzioni, evidenziazione
2	EX4	0,5	6 soluzioni, evidenziazione
3	EX4	0,75	4 soluzioni
4	EX4		6 soluzioni
1	EX5	0,3	evidenziazione, aiuto
2	EX5	0,7	solo evidenziazione
3	EX5		nessun aiuto

Figure 46: Excerpt of the table "livello\_difficolta"

Other tables exist, dedicated to decoding the indices used within the database (laterality, gender, province etc.). The common structure of this type of tables is composed of just 2 fields: the index and the corresponding description.

#### 4.1.3.2. Stimuli database

The first version of the stimuli database was not based on an ontology of concepts. The repository was built manually, using the structure that apparently best resembled the domain to be modeled: each table represented either a category or a relation type. This structure hosted more than 6000 stimuli (texts, images and sounds).

It featured a total of 84 tables, of which the ones listed below represent categories:

- *Clothes* (TAB\_ABBIGLIAMENTO);
- *Adjectives* (TAB\_AGGETTIVI);
- *Food* (TAB\_ALIMENTI);
- *Ambients* (TAB\_AMBIENTI);
- *Animals* (TAB\_ANIMALI);
- *Body actions* (TAB\_AZIONI\_CORPO);
- *Body parts* (TAB\_CORPO);
- *Economy* (TAB\_ECONOMIA);
- *Family* (TAB\_FAMIGLIA);
- *Geometric shapes* (TAB\_FIGURE\_GEOMETRICHE);
- *Uppercase letters* (TAB\_MAIUSCOLO);
- *Lowercase letters* (TAB\_LETTERE);
- *Places* (TAB\_LUOGHI);
- *Raw materials* (TAB\_MATERIE\_PRIME);

- *Means of transportation* (TAB\_MEZZI);
- *Nature* (TAB\_NATURA);
- *Digits* (TAB\_NUMERI);
- *Objects* (TAB\_OGGETTI);
- *Professions* (TAB\_PROFESSIONI);
- *School* (TAB\_SCUOLA);
- *Sports* (TAB\_SPORT);
- *Musical instruments* (TAB\_STRUMENTI\_MUSICALI);
- *Time* (TAB\_TEMPO);
- *Various* (TAB\_VARIE);
- *Verbs* (TAB\_VERBI).

Fields that are common to all the category-tables are:

- *lunghezza*, the number of syllables of the word;
- *numero*, indicating whether the word was singular or plural;
- *colore*, indicating the main color of the associated image stimulus;
- *stimulus*, the path of the image stimulus;
- *soundfile*, the path of the auditory stimulus associated to the database entry.

For what concerns the relations, the table called TAB\_RIASSUNTIVA explains which category tables are associated in each available relation. For example, considering the relation "indossa" (wears), the corresponding row of this table reads "TAB\_PROFESSIONI", "TAB\_INDOSSA", "TAB\_ABBIGLIAMENTO".

Beside this, several relation-table exist, that detail the single correspondences between records in different tables (e.g., the above mentioned "TAB\_INDOSSA").

#### **4.1.4. Problems with the first version**

The first version of CoRe presented several problems: first of all, despite the already mentioned ease-of-use of the tool, it was subject to frequent crashes- a particularly undesirable defect, especially for the type of software we mean to obtain, dedicated to impaired users for which this type of problem could be extremely detrimental. Besides, the correct function is guaranteed only with Windows XP. Trying to use the system under a Win 7 OS caused further crashes due to compatibility problems, among which issues of synchronization between the system clock and the software time.

## **4.2. System re-engineering**



CoRe was re-engineered in order to solve the aforementioned problems and to guarantee compatibility and portability, as well as to free the system from the bonds introduced by the use of proprietary software (e-Prime, MS Access).

For this reason the GUIs, software engine and exercises were rewritten in Java, while the databases have been re-implemented with MySQL.

#### 4.2.1. System architecture

The architecture of the system was improved, bringing the ontology into the schema (fig. 47). This brought substantial changes, especially from the point of view of the data storage. The stimuli database has in fact been rebuilt from scratch and optimized as described in chapter 4.2.2.

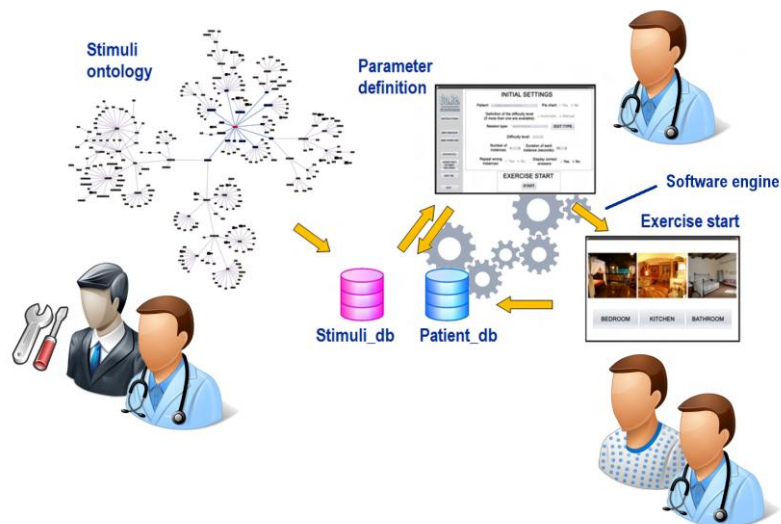


Figure 47: the structure of the re-engineered CoRe system

Despite having completely rewritten the entire system, from the visual point of view it has not been changed to avoid users potential disorientation and time losses due to the lack of familiarity with the interface. This is clearly visible in the new navigation diagram shown below.

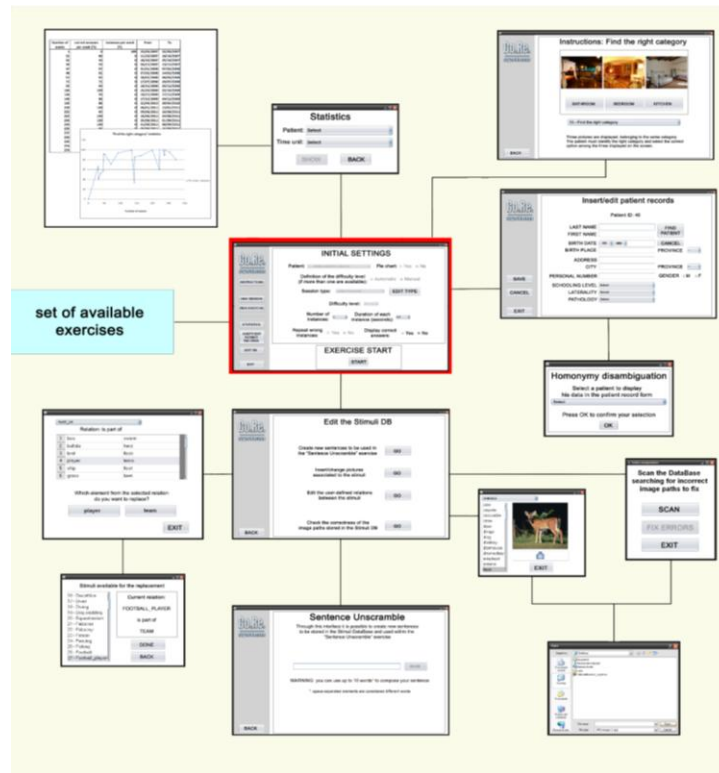


Figure 48: The new navigation diagram of CoRe

### 4.2.2. The databases

As described in chapter 3, the XML-SQL translator transforms the ontology into a raw database made of 2 tables (for concepts and relations). Of course, that is not the final form of the new repository. A further automated translation takes place, that analyzes all the entries in the raw tables and creates the ultimate, refined structure (see figure 49).

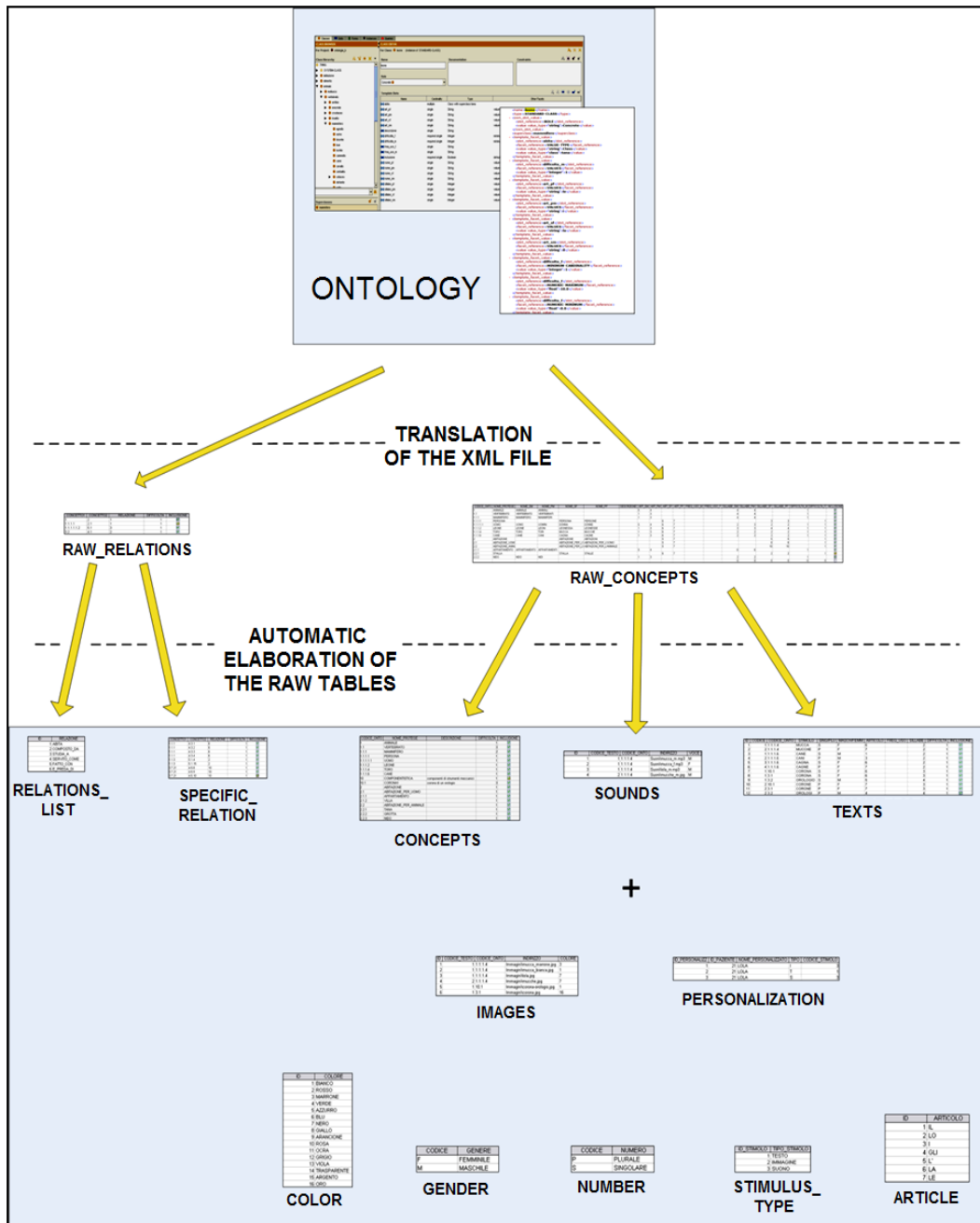


Figure 49: Schematization of the translation steps needed to transform the ontology into the new Stimuli database.

The raw tables are called "TAB\_CONCETTI\_TOTALE" and "TAB\_RELAZIONI\_ONTO". The first contains all the concepts from the ontology (and the previous stimuli\_db) organized in records whose fields mirror the attributes of the ontology.

- **CODICE\_ONTO:** contains the mnemonic code derived from the hierarchical structure of the ontology;

- NOME\_PROTEGE: the identifier used to define the concept within Protégé;
- NOME\_S: associated text stimulus (singular);
- NOME\_P: associated text stimulus (plural);
- DESCRIZIONE: description of the element;
- ART\_S: determinate article associated to the singular text stimulus;
- ART\_P: determinate article associated to the plural text stimulus;
- FREQ\_USO: word frequency of the text stimulus;
- SILLABE\_S: number of syllables of the singular text;
- SILLABE\_P: number of syllables of the plural text;
- DIFFICOLTA: empirical difficulty level associated to the concept;
- INCLUSIONE: inclusion flag.

CODICE_O	NOME_PROTEGE	NOME_S	NOME_P	DESC	ART_S	ART_P	FREQ	SILLABE_S	SILLABE_P	DIFFICOLTA	INCLUSIONE
1	ANIMALE	ANIMALE	ANIMALI		5	4		4	4	1	<input checked="" type="checkbox"/>
1.1	VERTEBRATO	VERTEBRATO	VERTEBRATI		1	3		4	4	1	<input checked="" type="checkbox"/>
1.1.1	MAMMIFERO	MAMMIFERO	MAMMIFERI		1	3		4	4	1	<input checked="" type="checkbox"/>
1.1.1.1	PERSONA	PERSONA	PERSONE		6	7		3	3	1	<input checked="" type="checkbox"/>
1.1.1.1.1	UOMO	UOMO	UOMINI		5	4		2	3	1	<input checked="" type="checkbox"/>
1.1.1.2	LEONE	LEONE	LEONI		1	3		3	3	1	<input checked="" type="checkbox"/>
1.1.1.4	TORO	TORO	TORI		1	3		2	2	1	<input checked="" type="checkbox"/>
1.1.1.6	CANE	CANE	CANI		1	3		2	2	1	<input checked="" type="checkbox"/>
2	ABITAZIONE	ABITAZIONE	ABITAZIONI		6	7		5	5	1	<input checked="" type="checkbox"/>
2.1	ABITAZIONE_UOMO	ABITAZIONE_UOMO	ABITAZIONI_UOMO		6	7		7	7	1	<input checked="" type="checkbox"/>
2.1.1	APPARTAMENTO	APPARTAMENTO	APPARTAMENTI		5	4		6	6	1	<input checked="" type="checkbox"/>
2.2	ABITAZIONE_ANIMALE	ABITAZIONE_ANIMALE	ABITAZIONI_ANIMALE		6	7		9	9	1	<input checked="" type="checkbox"/>
2.2.1	STALLA	STALLA	STALLE		6	7		2	2	1	<input checked="" type="checkbox"/>
2.2.2	NIDO	NIDO	NIDI		1	3		2	2	1	<input checked="" type="checkbox"/>

Figure 50: Excerpt of the table "TAB\_CONCETTI\_TOTALE"

The structure of "TAB\_RELAZIONI\_ONTO" is the following:

CONCETTO1	CONCETTO2	RELAZIONE	DIFFICOLTA	INCLUSIONE
1	2	1	1	<input checked="" type="checkbox"/>
1.1.1.1	2.1	1	1	<input checked="" type="checkbox"/>
1.1.1.1.1.2	5.1	3	1	<input checked="" type="checkbox"/>
3.2	4.1	2	1	<input checked="" type="checkbox"/>

Figure 51: Excerpt of the table "TAB\_RELAZIONI\_ONTO"

The table is composed of 5 fields:

- CONCETTO1: the identifier of the first element of the relation (from the field CODICE\_ONTO in TAB\_CONCETTI\_TOTALE);
- CONCETTO2: the identifier of the second element of the relation;
- RELAZIONE: the identifier of the relation;
- DIFFICOLTA: the difficulty level empirically attributed to the relation;
- INCLUSIONE: inclusion flag.

Other tables are associated to the two above, used to decode identifiers (COLORI, GENERE, NUMERO, LISTA\_RELAZIONI).

The raw database is optimized by automatically generating new tables: first of all, relations and restrictions contained in "TAB\_RELAZIONI\_ONTO" are examined and made explicit within table "TAB\_RELAZIONI\_SPECIFICHE". Once the population is completed, this table contains all the allowed relations between leaf-level concepts.

"TAB\_CONCETTI\_TOTALE" is split in two more tables, "TAB\_CONCETTI" and "TAB\_STIMOLI\_TESTO", represented below.

CODICE_ONTO	NOME_PROTEGE	DESCRIZIONE	DIFFICOLTA	INCLUSIONE
1	ANIMALE		1	<input checked="" type="checkbox"/>
1.1	VERTEBRATO		3	<input checked="" type="checkbox"/>
1.1.1	MAMMIFERO		2	<input checked="" type="checkbox"/>
1.1.1.1	PERSONA		1	<input checked="" type="checkbox"/>
1.1.1.1.1	UOMO		1	<input checked="" type="checkbox"/>
1.1.1.2	LEONE		1	<input checked="" type="checkbox"/>
1.1.1.4	TORO		1	<input checked="" type="checkbox"/>
1.1.1.6	CANE		1	<input checked="" type="checkbox"/>
10	COMPONENTISTICA	componenti di strumenti meccanici	1	<input checked="" type="checkbox"/>
10.1	CORONA1	corona di un orologio	3	<input checked="" type="checkbox"/>
2	ABITAZIONE		1	<input checked="" type="checkbox"/>
2.1	ABITAZIONE_PER_UOMO		1	<input checked="" type="checkbox"/>
2.1.1	APPARTAMENTO		1	<input checked="" type="checkbox"/>
2.1.2	VILLA		1	<input checked="" type="checkbox"/>
2.2	ABITAZIONE_PER_ANIMALE		1	<input checked="" type="checkbox"/>
2.2.1	TANA		1	<input checked="" type="checkbox"/>
2.2.2	GROTTA		1	<input checked="" type="checkbox"/>
2.2.3	NIDO		1	<input checked="" type="checkbox"/>

I	CODIC	CODICE_ONTO	STIMOLO	SING/PL	ARTICOLI	FREQ_USC	SILLABE	DIFFICOLT.	INCLUSIONE
1	1	1.1.1.5	MUCCA	S	6		2	1	<input checked="" type="checkbox"/>
2	2	1.1.1.5	MUCCHE	P	7		2	1	<input checked="" type="checkbox"/>
3	1	1.1.1.6	CANE	S	1		2	1	<input checked="" type="checkbox"/>
4	2	1.1.1.6	CANI	P	3		2	1	<input checked="" type="checkbox"/>
5	1	1.1.1.7	CAGNA	S	6		2	1	<input checked="" type="checkbox"/>
6	2	1.1.1.7	CAGNE	P	7		2	1	<input checked="" type="checkbox"/>
7	1	10.1	CORONA	S	6		3	1	<input checked="" type="checkbox"/>
8	1	3.1	CORONA	S	6		3	1	<input checked="" type="checkbox"/>
9	1	3.2	OROLOGIO	S	5		4	1	<input checked="" type="checkbox"/>
10	2	10.1	CORONE	P	7		3	1	<input checked="" type="checkbox"/>
11	2	3.1	CORONE	P	7		3	1	<input checked="" type="checkbox"/>
12	2	3.2	OROLOGI	P	4		4	1	<input checked="" type="checkbox"/>

Figure 52: Excerpt of the tables "TAB\_CONCETTI" and "TAB\_STIMOLI\_TESTO"

The correspondence between records from these tables is 1:2. From a single concept two different text stimuli can be created (its singular and plural instance)

Further two tables are created, called "TAB\_STIMOLI\_SUONO" and "TAB\_STIMOLI\_IMMAGINE". These associate audio files and images to elements from the table "TAB\_STIMOLI\_TESTO"

I	CODICE_TESTC	CODICE_ONTO	INDIRIZZO	VOC
1	1	1.1.1.5	Suoni\mucca_m.mp3	M
2	1	1.1.1.5	Suoni\mucca_f.mp3	F
3	1	1.1.1.5	Suoni\lola_m.mp3	M
4	2	1.1.1.5	Suoni\mucche_m.jpg	M

I	CODICE_TESTC	CODICE_ONTO	INDIRIZZO	COLORI
1	1	1.1.1.5	Immagine\mucca_marrone.jpg	3
2	1	1.1.1.5	Immagine\mucca_bianca.jpg	1
3	1	1.1.1.5	Immagine\lola.jpg	7
4	2	1.1.1.5	Immagine\mucche.jpg	7
5	1	10.1	Immagine\corona-orologio.jpg	1
6	1	3.1	Immagine\corona.jpg	16

Figure 53: Excerpt from the tables "TAB\_STIMOLI\_SUONO" and "TAB\_STIMOLI\_IMMAGINE"

Contrary to other tables, "TAB\_STIMOLI\_IMMAGINE" is not generated and populated automatically, because the information it contains cannot be inferred from the ontology.

#### 4.2.2.1. Database personalization

CoRe's personalizability reflects upon the database structure: given that for patients the identification of concepts is easier when the associated stimuli are familiar rather than general, the possibility exists to replace default generic stimuli with elements related to the patients' daily life (pictures of the family pets instead of generic stock photos of animals, names of familiars to identify family relationships- for example, the name "Pietro" instead of the word "grandson" etc.).

Within the database this personalization is implemented by associating the "proper name" of the element and the patient code (the personalized stimulus is only used for that specific patient) to generic concepts from "TAB\_CONCETTI", in order to create alternative textual, visual and auditory stimuli that replace the default ones during the execution.

The table "PERSONALIZZAZIONI" is made up of 5 fields:

- ID\_PERSONALIZZ, univocal record identifier;
- ID\_PAZIENTE, pointing to the table "Anagrafica" of the patient\_db
- NOME\_PERSONALIZZATO, with which the "alternative" concept is identified
- TIPO, text field indicating the stimulus type;

- CODICE\_STIMOLO, pointing to one of the 3 stimoli tables (“TAB\_STIMOLI\_IMMAGINE”, “TAB\_STIMOLI\_SUONO”, “TAB\_STIMOLI\_TESTO”)

ID_PERSONALIZZ	ID_PAZIENTE	NOME_PERSONALIZZATO	TIPO	CODICE_STIMOLO
1	21	LOLA	IMMAGINE	3
2	21	LOLA	TESTO	1
3	21	LOLA	SUONO	3

Figure 54: Excerpt of the table "PERSONALIZZAZIONI"

I	CODICE_TESTC	CODICE_ONTI	INDIRIZZO	COLORI
1	1	1.1.1.5	Immagini\mucca_marrone.jp	3
2	1	1.1.1.5	Immagini\mucca_bianca.jpg	1
3	1	1.1.1.5	Immagini\lola.jpg	7
4	2	1.1.1.5	Immagini\mucche.jpg	7
5	1	10.1	Immagini\corona-orologio.jp	1
a) 6	1	3.1	Immagini\corona.jpg	16

I	CODIC	CODICE ONTI	STIMOLO	SING/PL	ARTICOLI	FREQ_USC	SILLABE	DIFFICOLT	INCLUSION
1	1	1.1.1.5	MUCCA	S	6		2	1	<input checked="" type="checkbox"/>
2	2	1.1.1.5	MUCCHE	P	7		2	1	<input checked="" type="checkbox"/>
3	1	1.1.1.6	CANE	S	1		2	1	<input checked="" type="checkbox"/>
4	2	1.1.1.6	CANI	P	3		2	1	<input checked="" type="checkbox"/>
5	1	1.1.1.7	CAGNA	S	6		2	1	<input checked="" type="checkbox"/>
6	2	1.1.1.7	CAGNE	P	7		2	1	<input checked="" type="checkbox"/>
7	1	10.1	CORONA	S	6		3	1	<input checked="" type="checkbox"/>
8	1	3.1	CORONA	S	6		3	1	<input checked="" type="checkbox"/>
9	1	3.2	OROLOGIO	S	5		4	1	<input checked="" type="checkbox"/>
10	2	10.1	CORONE	P	7		3	1	<input checked="" type="checkbox"/>
11	2	3.1	CORONE	P	7		3	1	<input checked="" type="checkbox"/>
b) 12	2	3.2	OROLOGI	P	4		4	1	<input checked="" type="checkbox"/>

I	CODICE_TESTC	CODICE_ONTI	INDIRIZZO	VOC
1	1	1.1.1.5	Suoni\mucca_m.mp3	M
2	1	1.1.1.5	Suoni\mucca_f.mp3	F
3	1	1.1.1.5	Suoni\lola_m.mp3	M
c) 4	2	1.1.1.5	Suoni\mucche_m.jpg	M

Figure 55: Personalized records (in the stimoli tables) corresponding to the 3 records shown in the table "TAB\_PERSONALIZZAZIONI"

#### 4.2.2.2. Final structure

The ER diagram of the newly created database is reported below:

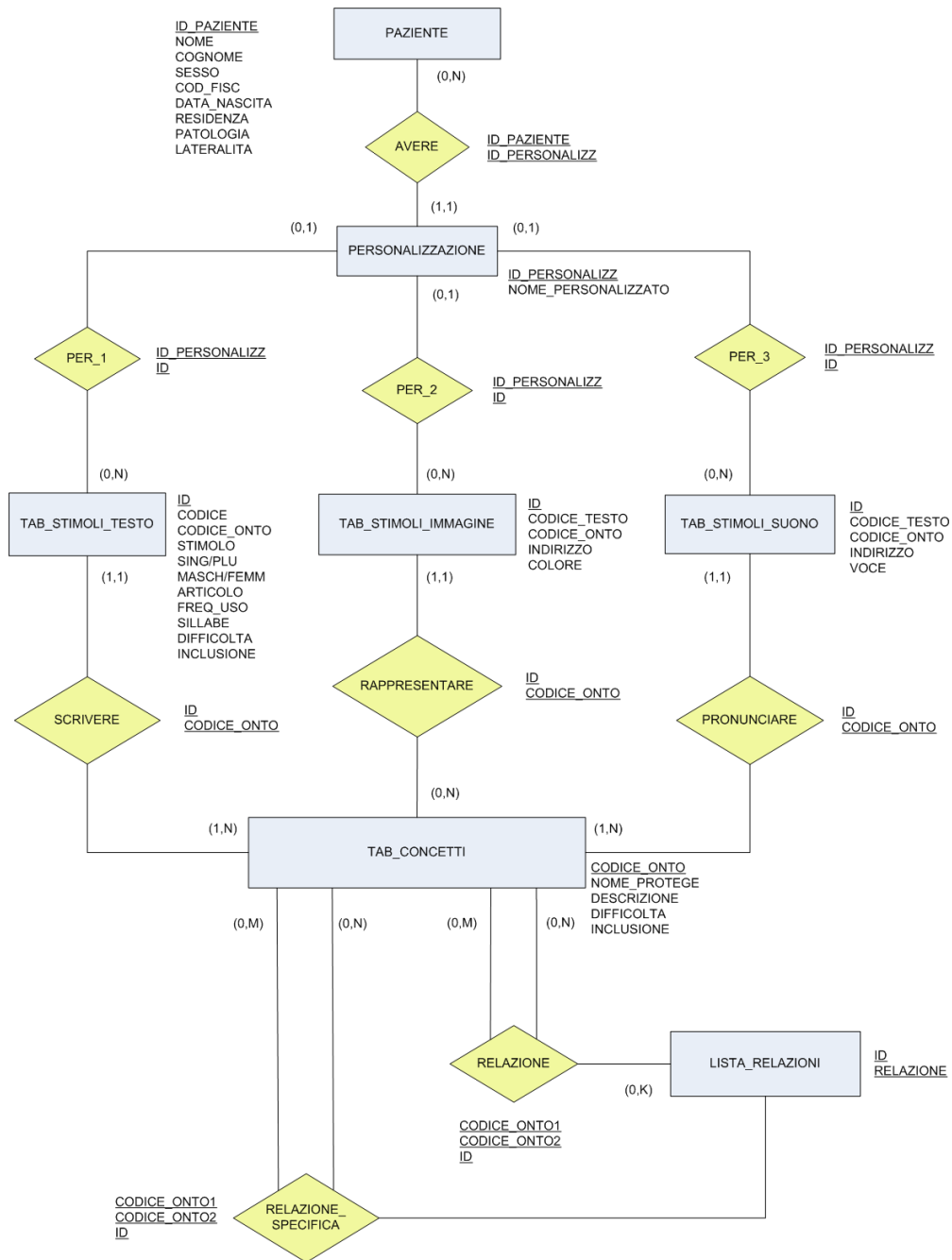


Figure 56: ER diagram of the new Stimuli\_db

### 4.2.3. The exercises

The 10 exercises implemented for rehabilitation of executive impairments (fig. 57) are:

- **word coupling**: eight words are displayed on screen. The patient must associate them in four couples, identifying the relations that exist between the stimuli;



- **pick the element:** a matrix of random text elements (letters or numbers) is displayed. The patient must identify and select the requested element;
- **find the intruder:** five words are displayed, four of which belong to the same category. The patient must identify the common category and consequently select the only intruder;
- **unscramble the sentence:** scrambled words are displayed. The patient must select them in the right order to form a sentence;
- **unscramble the images:** same as above, the patient must put the scrambled images in the right order to form a short story;
- **functional planning:** a verb describing an activity is displayed. The patient must select from a list the element related to that activity (e.g., if the action is "writing" and the elements are "candle", "doorknob", "pen" and "notebook", the patient will have to select the latter two);
- **image and sound:** a sound is played, whose duration can be either 'long' or 'short'. Meanwhile an image is displayed, whose dimensions can be 'big' or 'small'. The patient must evaluate whether duration and dimensions "match" (that is, big image + long sound or small image + short sound, according to a criterion explained before the execution) or not;
- **find the category:** three images are displayed. The patient must identify the common category to which all of them belong;
- **logical sequences:** a logical sequence of elements (numbers or images) with a blanked out element is displayed. The patient must identify, among several options, the correct one to complete the series;
- **logical analogies:** the textual version of a mathematical proportion is shown, with one of the four terms blanked out (e.g.: *FELINE* is related to *TIGER* as ? is related to *EAGLE*). The patient must identify the relation and therefore select the right element among the proposed options (in the proposed example, given the options *FISH* and *BIRD* the patient will select the latter).

The pilot study illustrated in chapter 6 is related to this set of exercises.

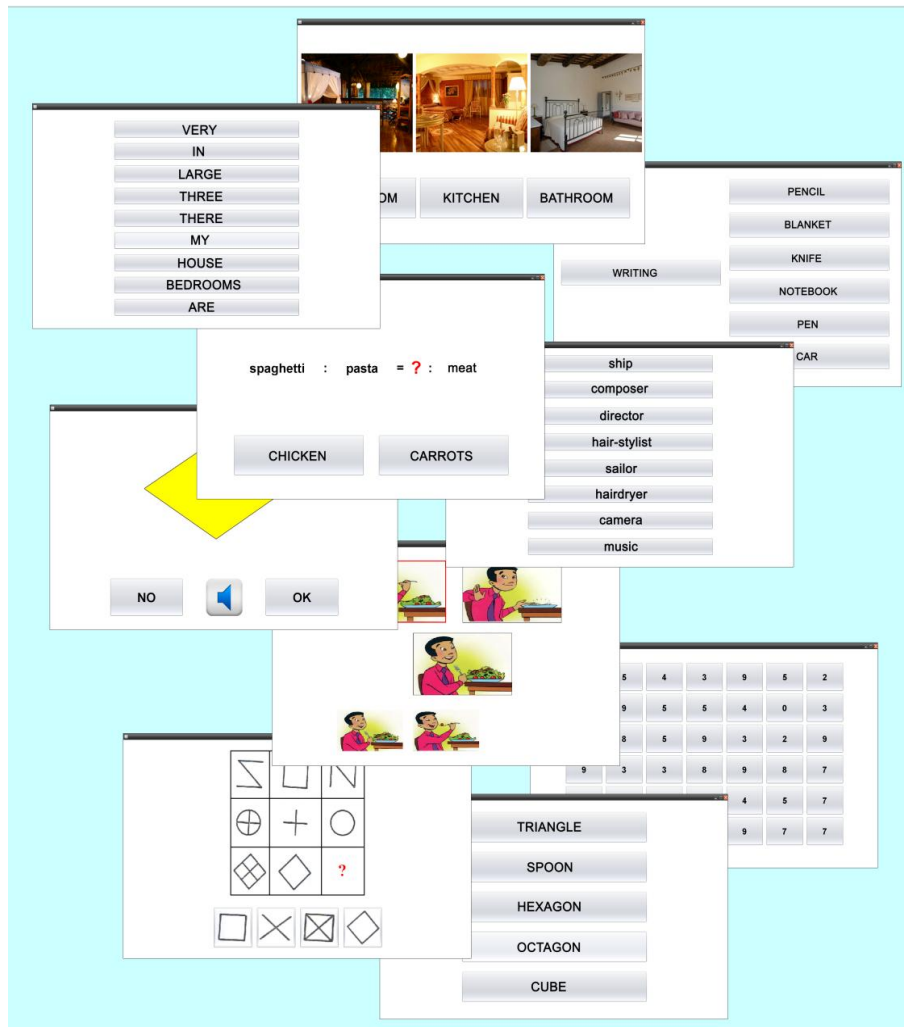


Figure 57: The 10 exercises currently available in the CoRe system.

Recently, in collaboration with speech therapists from S. Maugeri Foundation, we started developing new exercises for aphasic subjects, each focused on recovery of some of the specific cognitive areas described in Lichtheim's model (see chapter 1.1.4). The types which have been planned (and are currently being implemented) are the following:

- **phonemic discrimination:** two auditory stimuli are played in sequence and the patient must decide whether they are the same word/syllable;
- **visual and auditory coupling:** an auditory stimulus is played while a text is shown on screen. The patient must decide whether the text is the transcription of the spoken word/syllable or not;
- **visual/auditory lexical decision:** a stimulus is played/shown as a text on screen. The patient must decide whether it is a word or a nonword;

- **comprehension of spoken/written words:** a stimulus is played/shown as a text on screen, while 4 images are displayed: one represents the stimulus, one is a visual/semantic distractor, the others are wrong answers unrelated to the solution. The patient must identify the correct image;
- **comprehension of spoken/written sentences:** as above, but the stimuli are a description of a scene (the cue), and 4 images depicting similar situations. The patient must identify the one that represents the scene correctly;
- **discrimination of grammatical correctness of spoken/written sentences:** a sentence is spoken/displayed as text and the patient must decide whether it is grammatically and syntactically correct;
- **recognition of words/non-words (among other words/nonwords):** 4, 6, or 8 words/non words are spoken/displayed on screen; the player must recognize the requested one;
- **comprehension of oral/written definition:** the definition of an object is spoken/displayed on screen. Four answer options are displayed as text or images. The patient must select the right one;
- **classification:** an implementation similar to "3D find the intruder" is planned. An object is shown, together with some boxes, each of which contains some elements belonging to a common category. The player must place the object and put it in the correct box;
- **comprehension of spoken/written compound words,** according to rules of language;
- **recognition of syllables:** a matrix of syllables is shown. The patient must identify the requested one;
- **comprehension of spoken/written paragraphs;**
- **Daily life-oriented exercises for acalculia:** various exercises oriented to management of money (adding amounts, calculating change etc.). The sum of money to be handled will be represented either as text or images of the banknotes: this way it will also be possible to implement a task of recognition of banknotes (matching the image of the banknote with its value).

## 4.3. Additional features

### 4.3.1. Monitoring of stress and fatigue

Since CoRe lends itself to homecare applications, a function allowing the detection of patients' stress states and their influence on the performances is undoubtedly very useful.

In an attempt to satisfy this requirement, the integration of a body gateway device to the system has been attempted.

Even if we are aware that different sensors are needed for stress measurement, we performed a preliminary experiment with an ECG sensor, under the hypothesis that stress could increase heart frequency. In fact our first aim was to test the feasibility of a real-time alert to be provided to the patient in case of critical situations. Since this functionality was not the core of our system, we used an experimental sensor by ST Microelectronic, that the company provided to the laboratory within a collaboration agreement. Unfortunately, that sensor did not show good performances. However, since we think the idea is valuable, we illustrate our efforts in the following subsection.

#### 4.3.1.1. The device

The hardware chosen for the implementation of the application was an experimental sensor by ST Microelectronic (a first, raw version of their BodyGuardian device, fig. 58), supplied by the producer for research use.



Figure 58: the commercial version of the BodyGuardian by ST Microelectronics

Beside the device itself, the package provided by ST included:

- the charger
- the docking station for the charge
- some single-use sensorized patch, with integrated snap clip buttons to be attached to the device itself.

The device appeared to be suited for our purpose, due to its small dimension, ease of application and simplicity of use (the sensor, clipped to the patch, must be put on the patient's chest and just turned on by pressing a button), which mean a low impact on the wearer's daily activities. Besides, the interaction required from the user is extremely reduced, which also contributed in making that device suitable to our needs.

It allowed the detection of several vitals- ECG, heart rate, breath rate and activity level- and the signal detection/elaboration activities were based on proprietary algorithms. The wireless device worked as a 116 samples/sec holter device, automatically recording abnormal signals while

worn and active (it was provided with a 128MB internal memory); it needed to be periodically connected to a PC (via bluetooth) for the download of the stored data.

The fruition of the data recorded by the sensors happened with 2 modalities:

- "test mode", a real time visualization that showed the trace(s) during the recording, supposing that the device was connected to the PC (fig. 59);

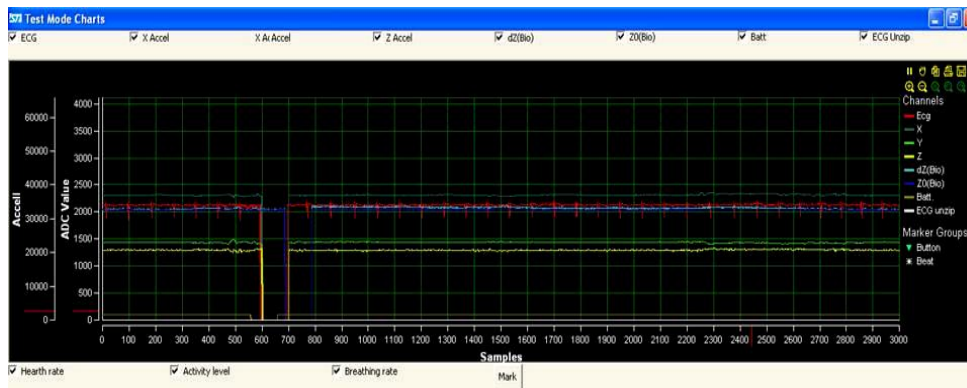


Figure 59: the real-time visualization of the sensed traces

- "logging mode", internal storage of values (by default when the device was active and placed on the patient, it was supposed to record 2-minute data streams at random intervals; plus, it also recorded vitals during critical events, that is whenever an abnormality was detected by the logic of the device) and subsequent download of all the data recorded (the device needed to be connected to a PC).

Since the automatic detection of critical events did not work (we purposely reproduced tachycardia events, making sure the internal activation thresholds were surpassed, but the recording did not start), we chose to exploit the "test mode" modality for integration in the system.

While charting the detected traces on screen, the "test mode" also copied the sample data row by row on a text file, which was kept open for writing until the disconnection of the device.

CoRe could not access that file while it was being edited, so we could not exploit it for implementation of real-time functions within the system. Our aim became the creation of a module to automatically analyze the data produced by the device, detect tachycardia and identify the exact event that caused it, *after* the end of the session.

To implement the monitoring function, the exercises needed to be slightly modified: the general structure of a task implies the visualization of a sequence of screens (welcome screen, instructions visualization,

execution, feedback etc), coupled with background elaboration of data based on the patient's interaction with the application. Considering that the sensor associated a timestamp to every sample it stored, if the goal was the "a posteriori" matching of stress state and stress cause, it was necessary to also apply a timestamp to every event of the exercise and not only to the execution screen (whose duration represents the instance's RT). The logical choice was to consider screen changes as events: this way, at every transition between screens, a record (event\_id, patient\_id, session\_id, event description, timestamp) was appended into the aptly created table "tempi" of our patient database.

Once the synchronization of the device with the PC had been completed, and the creation of its log file had been executed, the data was analyzed to look for tachycardia (detected analyzing the HR values provided by the device). If found, the corresponding timestamp was compared to those in the "tempi" table. This way it was possible to precisely indicate which event triggered the stress state.

This function, though asynchronous, could prove useful for several purposes- mainly, the refinement of the system, in order to "smooth out" the parts of the execution that appeared to cause stress in multiple patients.

Unfortunately, several problems emerged with the sensor, that made the integration impossible:

- the compatibility was not guaranteed for OSs other than 32bit Win XP. We tried the device on Win 7 and verified that while the "test mode" did not perform properly, failing at providing a continuous data stream, the logging mode completely stopped working. This made it necessary to use a virtualization of a WinXP OS (not the best solution for use by patients);
- the association procedure between the device and the PC (necessary for the communication of data via bluetooth) didn't work smoothly as expected: workarounds were necessary, which would be too intricate to be executed by a user; plus, for some reason, the association was not memorized correctly and had to be performed every time the sensor was used;
- while in theory the activation of the "test mode" only required a press of the device's ON-OFF button, we verified that this only worked about 50% of the times;
- as already mentioned, the automatic detection of critical events in "logging mode" did not work; plus, after the completion of a recording, a popup window would often appear onscreen, signaling a failure in the storing operation.

For all these reasons the stress monitoring function has currently been shelved, but as soon as a properly working sensor is available, it can be reintegrated in CoRe.

### 4.3.2. Weighted score

A weighted score (WS) was also created, that allows the comparison of different types of exercises, to make the assessment of actual long-term improvement in the patient's abilities possible.

There are 4 factors influencing the results of an exercise, and that, therefore, are taken into consideration to build the WS:

- exercise type.
- difficulty level of the execution;
- execution time;
- correctness of answer - the score is incremented in case of accuracy = 1. No points are assigned otherwise.

These are calculated individually and then summed to obtain the total WS.

The first 3 items of the list are explained below.

#### 4.3.2.1. Exercise type

Some exercises are intrinsically harder than others: for this reason, experimental data gathered during tests on healthy subjects, carried out at the Mondino hospital (see Chapter 6), have been analyzed. Based on these performance parameters, all the exercises have been ranked in order of their level of difficulty.

In particular, the parameters analyzed are:

- number of right/wrong/non given answers for each exercise, so that a higher difficulty score is assigned to types characterized by fewer correct responses;
- response time, following the assumption according to which longer RTs indicate harder exercises.

Table 2: Accuracy, "difficulty rate" and response times for every exercise type. The most and least difficult types are highlighted in red and green

EXERCISE TYPE	ACC1	ACC0	ACC2	DIFFIC RATE	AVG RT	STD_DEV RT
Find_intruder	172	11	2	0,08	8,15	9,60
Word_coupling	455	68	0	0,15	19,56	10,68
Pick_element	208	9	0	0,04	19,20	10,60
Image_sound	390	27	0	0,07	3,43	3,47
Functional_plan	432	15	1	0,04	5,46	3,49

Unscramble_sentence	424	103	1	0,25	16,73	14,14
Unscramble_images	120	68	2	0,58	22,43	17,46
Logical_sequences	219	137	9	0,67	17,96	13,61
Logical_analogies	361	9	0	0,02	6,06	4,56
Find_category	352	8	1	0,03	4,86	4,67

A more detailed table containing data related to patients' performance can be found in Appendix A (table A2).

It was thus revealed that *unscramble images* and *logical sequences* are the most complex exercises, showing the highest "difficulty rate" (that is, the quotient of the division of non-correct and correct answers) and particularly long execution times. On the other hand, *functional planning*, *logical analogies*, *find the category* and *image and sound* are easier: correctly solved most of the times, and with short RTs (see table 2).

The resulting WS (the type-related fraction) is shown in table 3.

Table 3: WS assigned to each exercise type based on test performances of healthy individuals.

EXERCISE TYPE	TYPE-RELATED WS
Unscramble images	25
Logical sequences	25
Word coupling	20
Unscramble sentences	20
Pick the element	18
Image and sound	15
Functional planning	5
Logical analogies	5
Find the category	5
Find the intruder	5

#### 4.3.2.2. Difficulty level

To define the fraction of WS concerning the difficulty level of the exercises, the operation is very simple. Considering the following table, where the numbers of difficulty levels available for each type are mapped, the score is obtained by simply dividing the dedicated fraction (25% of the total WS) in as many parts as the difficulty levels available and multiplying the result by the chosen level:



$$\frac{25}{levels_{available}} * level_{selected}$$

Table 4: Number of difficulty levels available for each type of exercise

<b>EXERCISE TYPE</b>	<b>NR OF DIFFICULTY LEVELS</b>
Word coupling	3
Unscramble images	2
Unscramble sentences	3
Pick the element	3
Functional planning	4
Logical analogies	2
Find the category	2
Image and sound	2
Logical sequences	2
Find the intruder	2

For example, execution of a level 2 word coupling would have a difficulty score of 16.66 (that is,  $25/3 * 2$ ).

#### 4.3.2.3. Execution time

To define this WS it is necessary to first execute the difference between Timeout time (TT, the maximum time allowed for the execution before CoRe times the exercise out) and response time (RT) to obtain the "saved" time and then set up a proportion in order to obtain the needed score.

$$x = \frac{TT - RT}{TT} * 100$$

$$WS = x * 0.25$$

Considering  $TT = 60000$  and  $RT = 10000$  (both expressed in milliseconds), the resulting WS would be  $(50000/60000 * 100 * 0.25) = 20.8$

#### **4.3.2.4. Results**

Elaborating data available in the table "sessioni" (it's possible to find an excerpt, also showing additional fields related to the components of the WS, in Appendix A, table A1), the total WS has been calculated for all the patients who underwent the complete treatment- first the WS related to every single instance and then the average WS of the specific sitting. The results of this elaboration are explained in detail in chapter 6.

---

# Chapter 5

---

## CoRe3D

AUTHOR'S NOTE: Content from papers submitted for publication [67][68] is reported in this chapter.

### 5.1. Reasons for the introduction of 3D features

In the latest years, with serious games gaining public attention and becoming an object of interest for researchers, 3D graphics have been introduced in many fields of the medical practice, among which psychological rehabilitation [43][44][45].

We chose to develop the visual version of the tasks in our system as 3D-based exercises for one main reason: for a human subject a direct interaction with actual, material objects is perceived as more natural than its "abstract" counterpart, that is interacting with bare text (the words that "represent" said objects).

Being the use of hundreds of physical items during rehabilitation sessions a clearly non-viable option, the best alternative is setting up an interaction between the user and a visual representation of every object.

This, of course, could be achieved using simple .jpg images easily available online, but the advantages the use of 3D models entails are clear: first of all, it allows a more complex interaction, giving users the chance to have a 360° view of the solids and, in general, to manipulate them at pleasure: secondly, the programmer has complete control over the looks of a model- every single vertex, edge and face of a solid is editable in any moment- something that does not happen with 2D images.

Before switching to the implementation stage of the 3D-based application, research activity had to be performed to gather background information that could back our hypotheses of feasibility up.

## 5.2. Background

Several studies have been performed in order to evaluate the potential advantages entailed by the introduction of a third dimension in applications for which the 2D approach has always been well-established; in particular, comparisons have been made between 2D and 3D approach in the development of document management systems.

While studies by Tavanti et al. [46] acknowledged benefits for the users' spatial memory, supposedly related to the interaction with a 3D environment, further experiments by Cockburn et al. [47][48][49] countered said conclusions, stating that, according to their results, beside the personal preference of the "cooler" 3D interface expressed by the users, no actual benefits come from the introduction of width in a document management system. If anything, confusion arises, due to the additional flexibility provided by the third dimension.

Regarding the learning field, Dalgarno et al. [50] state that 3D virtual environments have unique characteristics that can benefit learning more than 2D applications: to the user, representational fidelity, immediacy of control and presence translate into a higher level of participation (thus leading to an increase in motivation and compliance), and a higher transfer of knowledge and skills to real situations.

As it can be seen, no stringent conclusion can be drawn based on this source material. Darken et al. [51] define a taxonomy of exercises based on the interactive tasks involved. After submitting said exercises to 27 healthy volunteers, they are able to infer a standard for predicting the usefulness of 3D technology: the results of their tests clearly suggest that anytime a spatial task (the ones defined as "spatial selection", "position", "orientation" and "path" in their classification) is performed, 3D Virtual Environments improve the users' performances. On the contrary, if a task does not require spatial interaction ("non-spatial selection", "reading/writing text" and "quantification") switching from a 2D to a 3D environment impacts badly on both the error rate and the task completion time.

In the year 2000 a guideline has also been created by Baldonado et al. [52], including eight rules to support the developer in the creation of a multiple-views system for information visualization.

Since then several applications have been created, that fully support mixed 2D/3D interfaces (Baumgartner [53], Bellotti [54]), thus showing how the integration is a winning strategy compared to choosing one approach and "forcing" it over task types for which a different implementation approach would turn out to be more user-friendly, affordable and efficient.

Regarding navigation in virtual environments, which is the field of interest for us, further analyses have been performed through the years:

guidelines can be derived from the results of these researches which can prove very useful in the development phase.

First of all, Vidal et al. [55][56] found out that allowing pitch rotation of the camera during their tests not only did not help the navigation, but also caused confusion in the users.

Based on the analysis of parameters acquired, as well as on direct feedback by the volunteers who accepted to undergo their tests, Burigat et al. [57] stated the importance of navigation aids during 3D navigation tasks. Minocha et al. [58] were able to identify the design choices that most negatively impact on the users' experience, as well as to define design guidelines to prevent/correct such errors.

Other aspects have to be taken into account while creating a navigable 3D virtual environment- mainly, the gender and age of the target user.

In fact several studies [59][60][61] agree in asserting that basic differences exist between male and female spatial perception: while women rely predominantly on landmark cues, men use both geometric and landmark cues. These findings stress the need to prevent biasing of environmental construction towards one strategy/cue type over another, in order not to create exercises with different cognitive load for different types of users.

Beside the gender, another demographic characteristic that dramatically influences the fruition of 3D exercises is age: very predictably, elderly subjects have different sensory/perceptual processes compared to young adults, which represent the usual target for 3D applications. Elderly-oriented navigational tasks must thus provide an interaction paradigm that keeps sensory acuity problems (typical in elderly subjects) into account, "adapting" itself to the user's abilities. While taking care of this problem, the design of the application must be as elementary and user-friendly as possible.

Lastly, the characteristic that probably matters the most, considering the goal of our study (that is, the creation of 3D Virtual Environments for cognitive rehabilitation), is the difference between the cognitive mechanisms of healthy and brain-injured users.

According to Livingstone et al. [62], participants with moderate to severe traumatic brain injury (TBI) are as capable as healthy participants at navigating in the presence of clear proximal landmarks, but impaired when required to navigate without proximal landmarks. Besides, TBI survivors have difficulty constructing, remembering or using cognitive maps of large-scale space during spatial navigation. That is, TBI survivors have a deficit in cognitive mapping ability.

There are some important abilities that are spared after TBI and those contribute to way finding, namely the abilities to associate a landmark with a destination, to discriminate that landmark from other distracting stimuli, and to navigate to a destination using that landmark.

## 5.3. Blender

Blender is the tool we used to develop our 3D exercises; it's a free multiplatform software for modeling, rigging, animation, compositing and rendering of 3D content.

The definition available on the official site is reported below:

*"Blender was first conceived in December 1993 and became a usable product in August 1994 as an integrated application that enables the creation of a diverse range of 2D and 3D content. Blender provides a broad spectrum of modeling, texturing, lighting, animation and video post-processing functionality in one package. Through its open architecture, Blender provides cross-platform interoperability, extensibility, an incredibly small footprint, and a tightly integrated workflow. Blender is one of the most popular Open Source 3D graphics applications in the world.*

*Aimed at media professionals and artists world-wide, Blender can be used to create 3D visualizations and still images, as well as broadcast- and cinema-quality videos, while the incorporation of a real-time 3D engine allows for the creation of 3D interactive content for stand-alone playback or video games.*

*Key Features:*

- *Fully integrated creation suite, offering a broad range of essential tools for the creation of 3D content, including modeling, uv mapping, texturing, rigging, skinning, animation, particle and other simulation, scripting, rendering, compositing, post-production, and game creation;*
- *Cross platform, with an OpenGL GUI that is uniform on all platforms (customizable with python scripts), ready to use for all current versions of Windows (XP, Vista, 7), Linux, OS X, FreeBSD, Sun and numerous other operating systems;*
- *High quality 3D architecture enabling fast and efficient creation work-flow;*
- *More than 200,000 downloads of each release (users) worldwide;*
- *User community support by forums for questions, answers, and critique at <http://BlenderArtists.org> and news services at <http://BlenderNation.com>;*
- *Small executable size, easy distribution." [63]*

Its flexibility and ability to manage every step of the creation of graphics makes Blender the perfect candidate for the implementation of 3D-based features within our system [64].

The user interface may appear as not particularly user-friendly, given the necessity to fit a great number of editing tools and options in the workspace. Creating a complete user's guide for the software is beyond the scope of this text, but a short description of the main components used to create CoRe's 3D tasks can be useful to better understand the implementation stage: for this reason a discussion of said components is reported below.

### 5.3.1. Main GUI

Shown below is the main GUI of Blender. From this interface it is possible to control every aspect related to the objects' structure.



Figure 60: Blender's main GUI. In its *default* configuration it is made up of 4 windows: the 3D view (a), the Timeline (b), the Outliner (c) and the Properties (d)

The *default* configuration is made up of 4 components, each occupying a portion of the screen:

- the *3D view* (fig. 60, a), that allows the visualization of the model in the 3D space and from which it is possible to edit vertices, edges and faces of every object (mesh) in the scene, as well as to perform several types of transformations (translation, rescaling, deletion, duplication etc.) available as options in the "Object tools" tab (XX, left column);
- the *Timeline* (60, b), necessary to define keyframes during the creation of animations;

- the *Outliner* (60, c), from where it is possible to quickly select and manage every component featured in the tree-like representation of the scene;
- the *Properties* (60, d), a multi-tab window that gives full control over the characteristics of the virtual environment and the nature of the objects: materials, colors, textures, lighting, physical behavior can all be edited from this menu.

The first stage of the implementation, consisting in the modeling of the virtual environment and the meshes, has been carried out mainly from this interface.

Another selectable configuration, essential for the second stage of the implementation (that is, the definition of the objects' behavior), is the one called *game logic* (fig. 61).

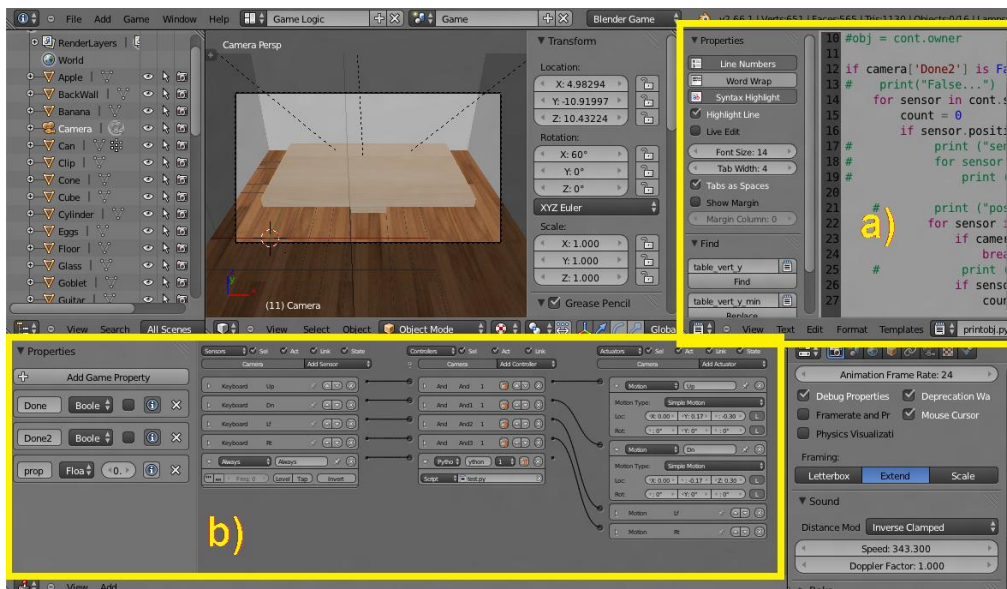


Figure 61: the game logic configuration of Blender's GUI. The text editor (a) and the Logic editor (b) are highlighted.

The main difference between this and the one described above is the presence of new windows:

- the *Text editor* (61 a), to allow the creation of python script to be integrated in the execution of the .blend file;
- the *Logic editor* (61 b), useful to define triggers and event-based execution of routines.

The latter is a particularly interesting component, that allows the developer to define complex behaviors and link them to specific events.

Figure 62 shows the structure of the Logic editor.



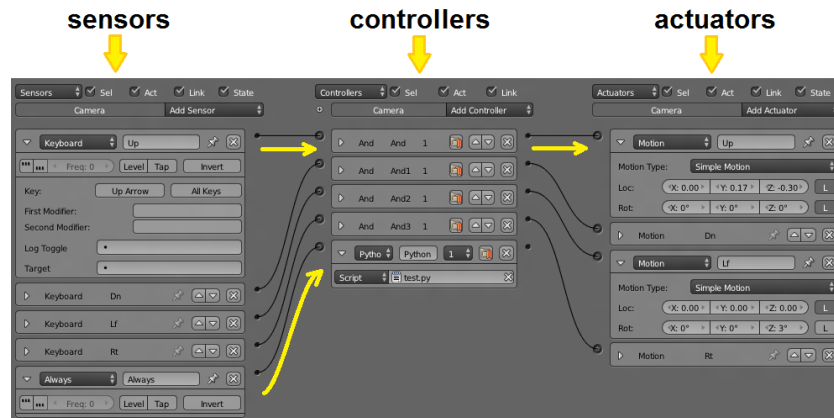


Figure 62: Detail of the Logic editor. Sensors (left), controllers (center) and actuators (right).

It is possible to define the behavior of every component just by connecting instances of the three main components:

- *sensors* (62, left), which can be setup to listen to specific events;
- *actuators* (62, right), used to indicate the action triggered by the listened event: a set of common action types is provided by default: among the selectable options scene switching, object motion/rotation, visibility change and property setting can be found;
- *controllers* (62, center), which implement logic operators (AND, OR, NOR...) and are used to combine multiple instances of sensors and/or actuators. Beside creating logic operations, the controllers also allow event-triggered activation of python scripts (or even specific routines within a script). This feature allows the implementation of complex functions which are not available by default.

In figure 62 two cases are highlighted with yellow arrows: the first one (top) connects a "keyboard" sensor that listens to keypress event- in this case, the UP arrow- to an actuator that moves the associated object (the camera) forward on the y axis. Basically, this sequence represents 1/4 of the programming needed to implement a basic mechanism of first-person ambient navigation.

The second case (fig 62, bottom) is an example of event-based script activation: being connected to an "always" sensor means that the "test.py" script is active since the start of the exercise. The content of this script is explained in chapter 5.5 and analyzed in Appendix B.

## 5.4. Implementation strategy

Before starting the implementation of 3D exercises, some important matters had to be taken into consideration:

- First and most important, during the creation of object models a compromise had to be reached between detail and complexity: the more the vertices composing the solid, the bigger the resulting object file and the computational power needed to render the scene. This will be even more important when the time comes to use CoRe and the re-engineered 3D exercises for telerehabilitation sessions: our typical cognitively impaired patients, that is mostly elderly people with very limited familiarity with modern technologies, very likely do not own powerful PCs. Thus it was necessary to keep the system requirements of our software as low as possible. As a thumb rule, we chose to set the maximum number of vertices-per-object to about 10.000.
- Secondly, any 3D exercise features a scenario that, as simple and limited in space as it may be, the user should be allowed to navigate freely, to get as close to the foreground objects as possible. This, together with the possibility to have a 360° view of the single objects on which the task is focused, implies the need of a "complex" control system: it is clear how the combination of mouse buttons and direction arrows cannot be able to guarantee the needed degree of interaction, and introducing more keyboard keys to the control system may well be considered way too complicated for our target users. For this reason another compromise was needed, this time between the degrees of freedom allowed to the user and the complexity of the controls.

## 5.5. The exercises

Among its "text-only" tasks, CoRe features the previously described exercise called "Find the intruder", that exploits the relations between the concepts represented in the ontology.

Despite being a very simple exercise that requires only one click to be solved, its appearance (5 words written on the labels of just as many buttons visualized on a white screen) is not very enticing for patients. This could be a cause of the drop in compliance we mean to avoid.

The strategy we chose to solve this problem consisted in transforming "Find the intruder" in a visual exercise: contrary to "Unscramble the sentence", in which grammar and syntax play a vital role, this exercise is based on recognition of concepts and thus better suited for re-engineering.

The result of the implementation, carried out in collaboration with the Computer Graphics Division of the Polytechnical University of Catalonia (UPC) is shown in fig. 63: a room with 3 bare walls and a floor with a big table as the only piece of furniture, where the 5 objects are placed.



Figure 63: The 3D-based visual implementation of the re-engineered "Find the intruder" exercise

To achieve this "randomness" in the placement, a Python [65] algorithm for scaling and positioning the objects has been implemented, which can be defined as a variation of a first fit 2d-packing algorithm [66] which is explained in Appendix B.

Different versions of the same 3D task have been implemented, to meet the needs of patients with different degrees of impairment.

Beside the "basic" version shown above, with the five objects scattered randomly on the table, another, apparently more complex one was implemented. Here, the elements are not immediately visible, but hidden into five closed boxes lined up on the table; in order to successfully complete the task, the user needs to open every box first, and see what's inside.

Since the introduction of the boxes increased the difficulty level of the exercise (and, consequently, the complexity of the control system required), the need to simplify some aspects of the execution emerged. This was done in order to avoid the introduction of unwanted complications that could interfere with the achievement of the main task.

The very idea of introducing boxes in the exercise made it necessary to choose the right approach to one particular issue. As mentioned above, our stimuli database is very wide and made up of elements belonging to diverse domains; it is clear that the difference between the objects' orders of magnitude can be substantial, and this can represent a problem. Let's say, for example, that the two categories are 'means of transport' (a car, a bike, a motorbike and a truck) and 'sports equipment' (a soccer ball): of course,

simply "wrapping" rectangular boxes around the objects would render the exercise useless, because the size difference itself would basically give away the solution to the task: for this reason, boxes with fixed size and shape are used, and a scaling algorithm has been created to fit all the objects in.

A choice needed to be made between maintaining a constant scale for all the objects (that is, keeping the size difference realistic in order to contextualize the exercise and help the patient identifying the categories at the price of a potentially significant loss in objects' visibility- see the example of truck/car etc. vs. soccer ball) and scaling every object down/up to fit the box size (losing the information about the size difference, but guaranteeing the recognisability at a glance of all the objects).

The second solution was chosen (Figure 64 c-d), based on the fact that the main acceptability criterion for any 2D image is the visibility of the subject and no size difference information are ever available in picture-based exercises.

At that point, the only apparently trivial problem of the interaction with the boxes had to be tackled; while the initial intention was to develop a mouse-based drag-and-drop type of command, that would have required the user to simulate the gesture of lifting the lid in order to open the box, the simplification led to the choice of a much simpler approach: a click on a box triggers a default animation of its lid lifting and moving out of sight.

To pursue the simplification mentioned above, the free roaming feature, available in the "no-box version" of the task, has been disabled here: at the start of the game the camera is placed in front of the table, so that all the boxes are within the field of view at the same time, and all the patient is required to do to see what's inside a box is click on it.

This selection, as already mentioned, triggers a default animation (Figure 65): while the lid lifts, the camera zooms to the selected box and tilts down to show the content. After a few seconds of wait, the camera zooms out and back to the starting position. The player is allowed to look into the boxes multiple times to re-examine objects already seen (a counter will keep track of the total number of views for each box). After that, he can complete the task by selecting the solution with the press of one of five aptly assigned keys within the keyboard.

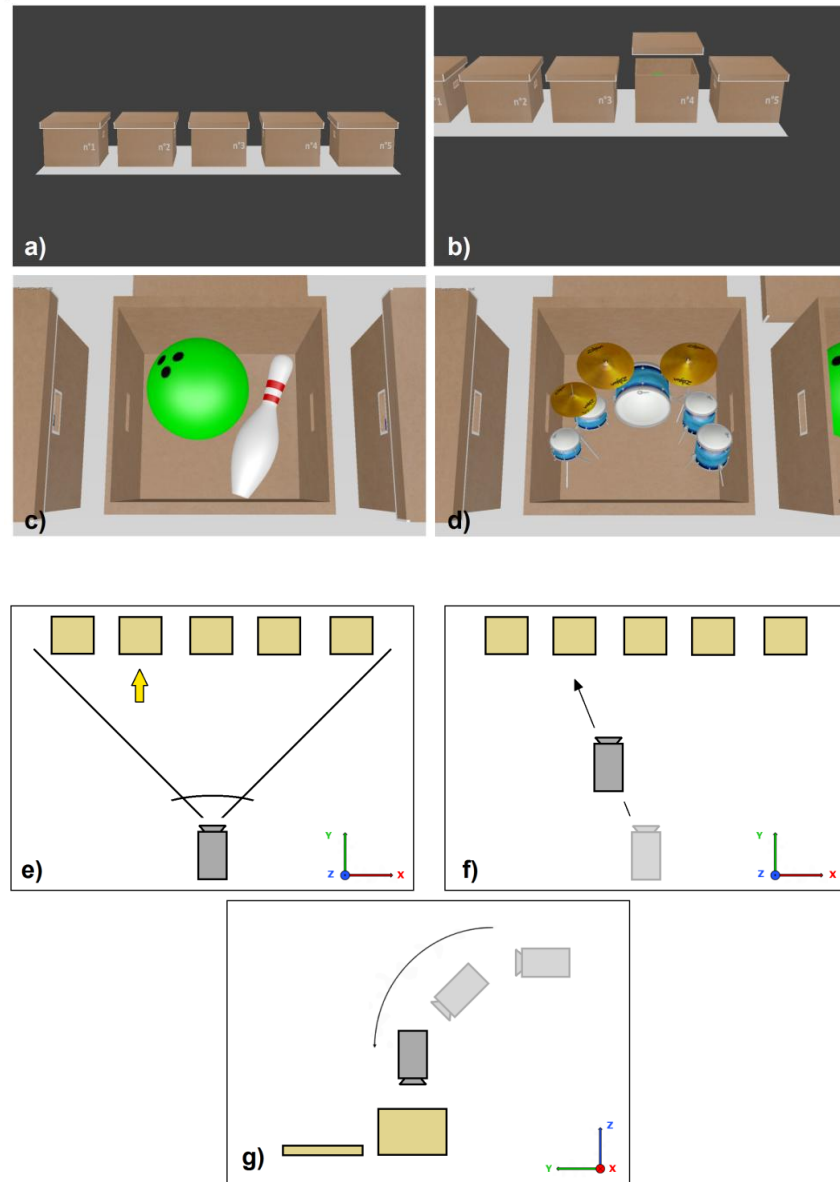


Figure 64: Camera motion during the automated animation in the "box version" of "Find the Intruder": once a box is selected (a), the lid opens (b) and the camera zooms in and tilts down over the box (c). Besides, (c) vs. (d) show how all the objects are scaled to fit the dimensions of the boxes. The camera positions and movements during the animation are schematized in (e), (f) and (g).

A very important issue was brought to light by the therapists during the analysis of this implementation schema: some of the patients could be afflicted with visual-spatial processing deficits, and the camera motions featured in the exercise could cause confusion and disorientation. This is why an alternative version of the task was implemented: here, when the player clicks on the box, the camera moves parallel to the table until it stops in front of the selected box. The lid still opens automatically, but there is neither forward motion nor tilt of the camera. Instead, the object

flies out of the box and then towards the player; It hovers in front of the camera, slowly spinning on its vertical axis for a few seconds to allow a better view, and then goes back into the box.

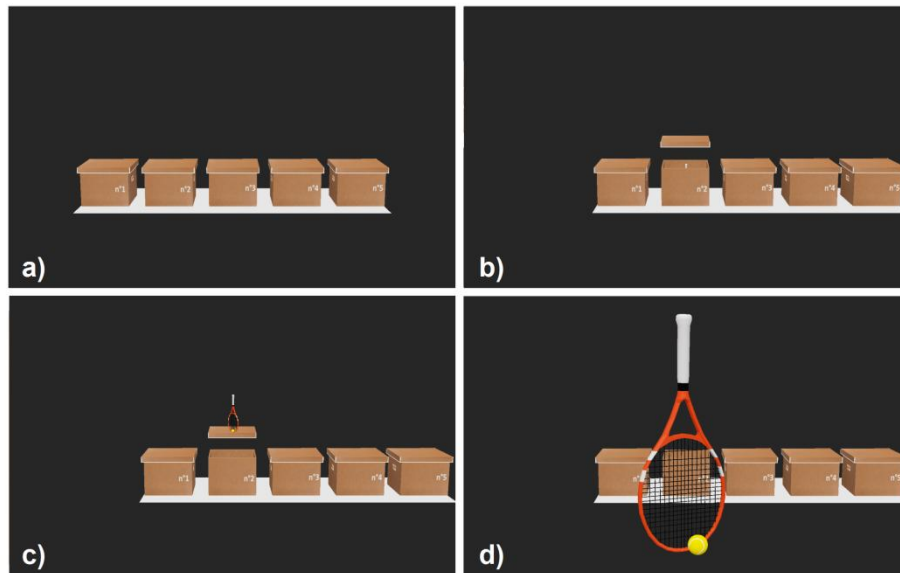


Figure 65: Camera motion during the automated animation in the alternative "box version" of "Find the Intruder": once a box is selected (a), the lid opens (b) and the object flies towards the camera (c) and rotates on its vertical axis for a few seconds (d) before flying back into the box.

---

# Chapter 6

---

## CoRe validation

Once the development was completed, the first version of CoRe has undergone a test phase to verify the overall function of the system as well as to assess its usability.

### 6.1. Validation on healthy volunteers

#### 6.1.1. Experimental setup

The test was performed at the Mondino Foundation's Alzheimer Evaluation Unit on a group of healthy volunteers, heterogeneous from the point of view of age, education and familiarity with a PC, who underwent a simulated rehabilitation sitting in presence of a therapist for support.

- The sample included 38 subjects, 20 above and 18 below the 60-years threshold.
- The level of education was to be expressed by choosing an option among the ones made available: "<=5 years", "5 to 8 years", "13 to 18 years", ">18 years". This subdivision was chosen based on the Italian education system, according to which educational cycles are structured as follows:
  - 5-year primary/elementary education;
  - 3-year first grade secondary (middle school) education;
  - 5-year second grade secondary (high school) education;
  - 3-year (at least) post-secondary education (university).

- The degree of familiarity with a PC was also discretized in 5 classes: "none", "almost none", "average", "good" and "excellent", to be picked based on self-assessment by the subject.
- A fixed protocol was created for the experimentation, requiring every subject to undergo the same set of exercises (some instances from every exercise type and every available difficulty level).

With the usability evaluation as a main goal for this stage, it was decided to request active feedback from the users at the end of the test rather than relying on elaboration of performance data. For this purpose, our choice fell on the standard questionnaire based on SUS, which is the most widespread and reliable usability evaluation tool currently available.

The SUS questionnaire is a Likert scale made up of 10 items, useful for global subjective assessment of usability. The scoring is provided by expressing one's agreement with the statement expressed in the item- that is, selecting one of the 5 options provided, ranging from "completely disagree" to "completely agree", with the 3rd level expressing neutrality about the subject concerned.

Here is a complete list of the items in the questionnaire:



	Strongly disagree					Strongly agree
1. I think that I would like to use this system frequently	1	2	3	4	5	
2. I found the system unnecessarily complex	1	2	3	4	5	
3. I thought the system was easy to use	1	2	3	4	5	
4. I think that I would need the support of a technical person to be able to use this system	1	2	3	4	5	
5. I found the various functions in this system were well integrated	1	2	3	4	5	
6. I thought there was too much inconsistency in this system	1	2	3	4	5	
7. I would imagine that most people would learn to use this system very quickly	1	2	3	4	5	
8. I found the system very cumbersome to use	1	2	3	4	5	
9. I felt very confident using the system	1	2	3	4	5	
10. I needed to learn a lot of things before I could get going with this system	1	2	3	4	5	

The SUS score is calculated as follows: first the score contributions from each item (ranging from 0 to 4) are summed: for items 1,3,5,7, and 9 the score contribution is the scale position minus 1. For items 2,4,6,8 and 10, the contribution is 5 minus the scale position. To obtain the overall value of SU, the sum of the scores must be multiplied by 2.5.

Since SUS value is not a percentage expression of "how usable the software is" (despite being expressed by a 0-100 value), 50% is not considered the threshold of "acceptability". Research by J.Sauro [1] contributed in defining a reliable metric to assess the results of a SUS score evaluation: given the sample sets chosen by the author for his research, the following representations are derived.

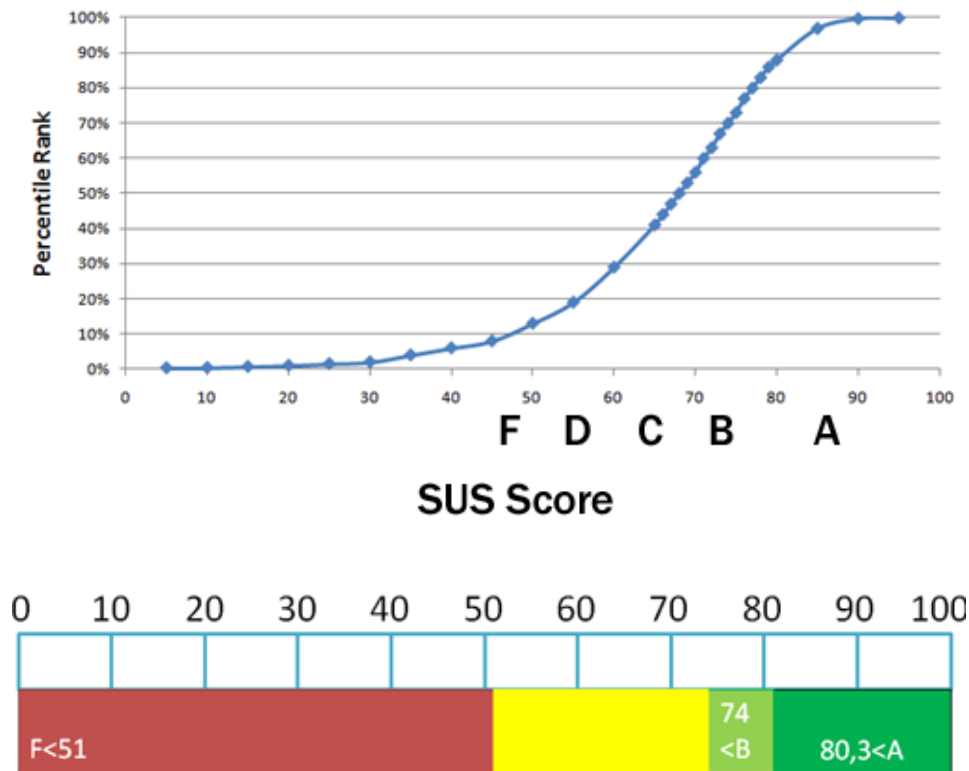


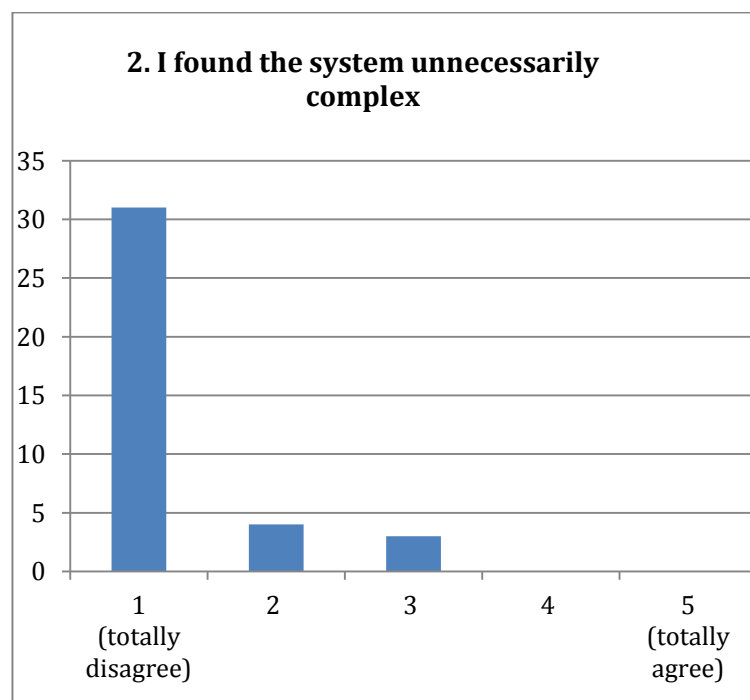
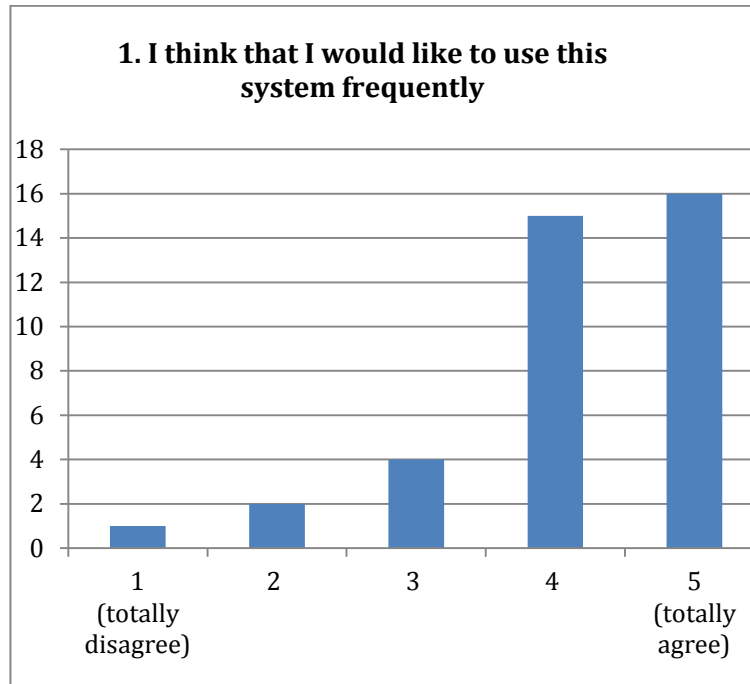
Figure 66: equivalent representations of metrics for the assessment of SUS score according to research by J. Sauro. Ranks below 51 correspond to very poor usability, while grades over 80 rank a system as top-class.

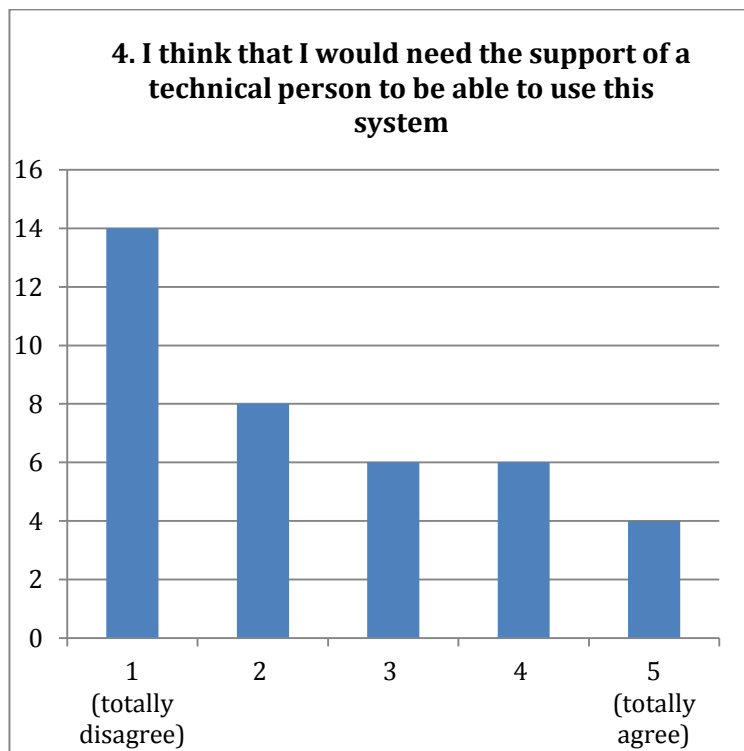
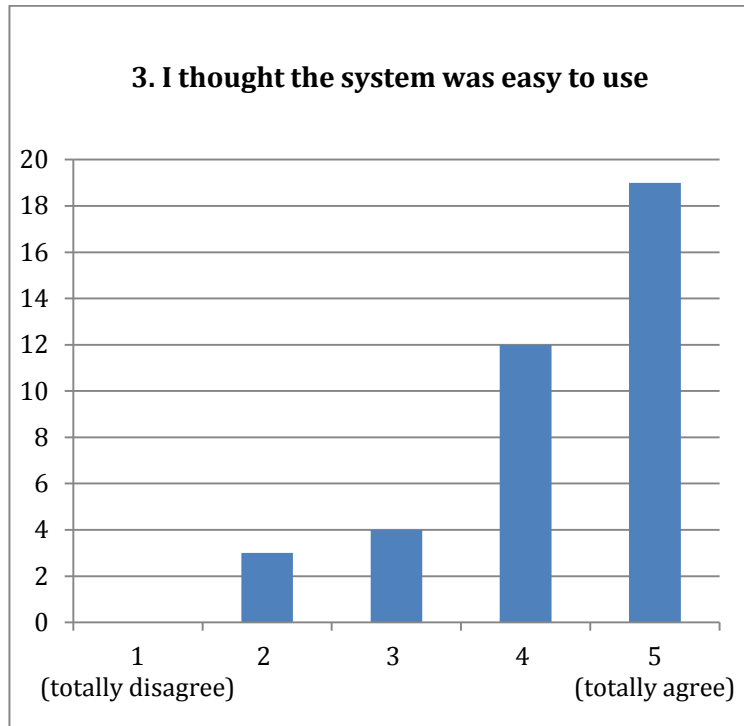
For example, a raw SUS score of a 74 converts to a percentile rank of 70%, meaning that the system has higher perceived usability than 70% of all products tested. It can be interpreted as a grade B.

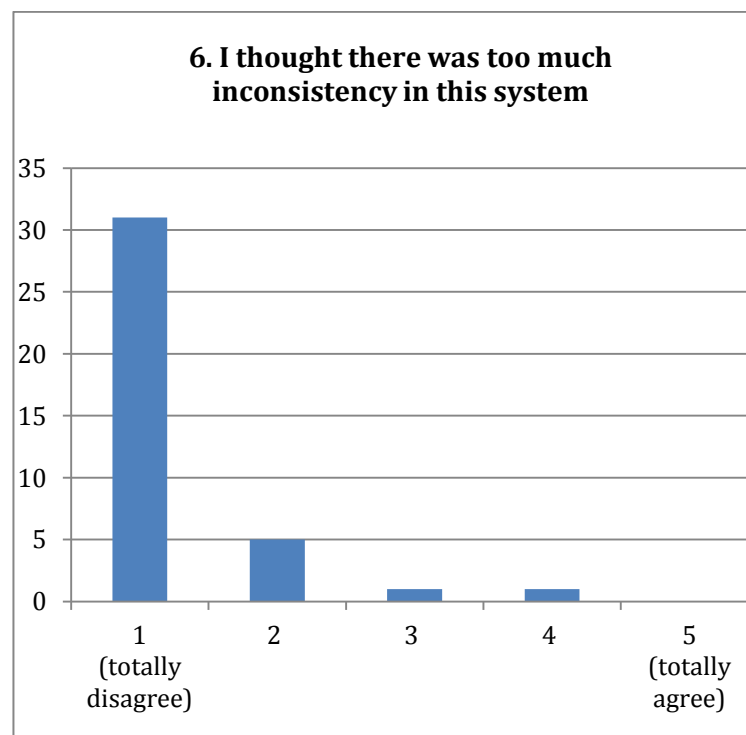
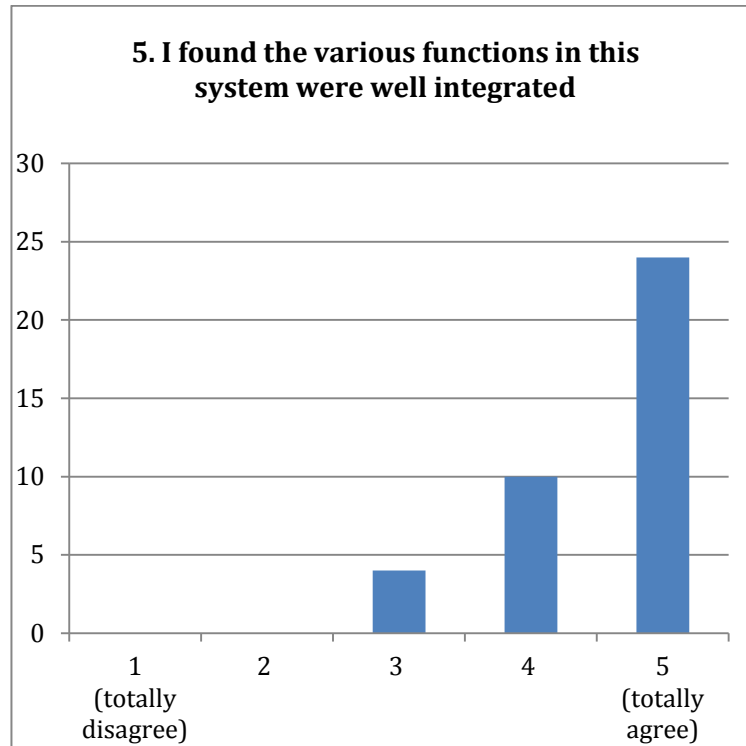
It's necessary to score above an 80.3 to get an A (the top 10% of scores). A score of 68 corresponds to a C and anything below a 51 is an F (bottom 15%).

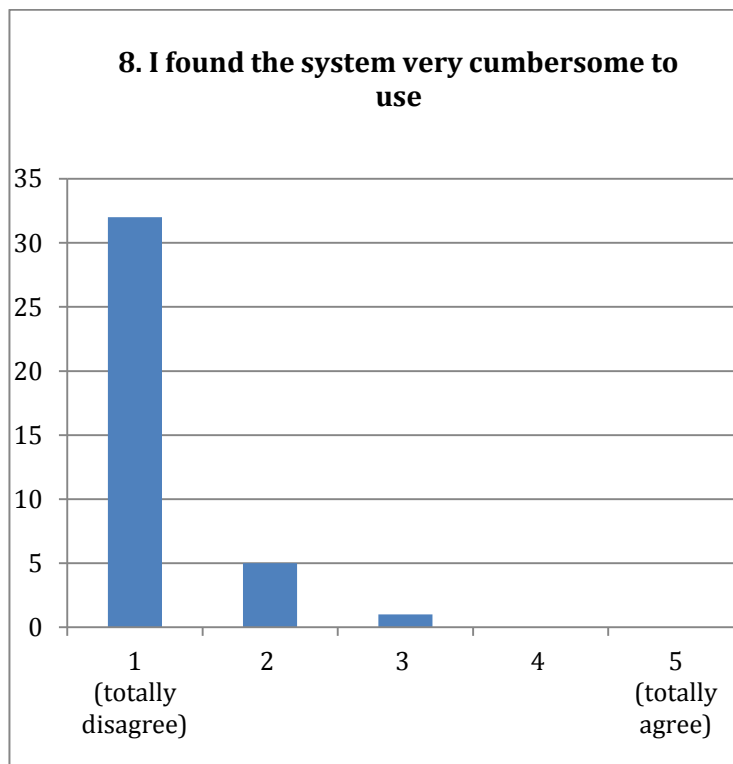
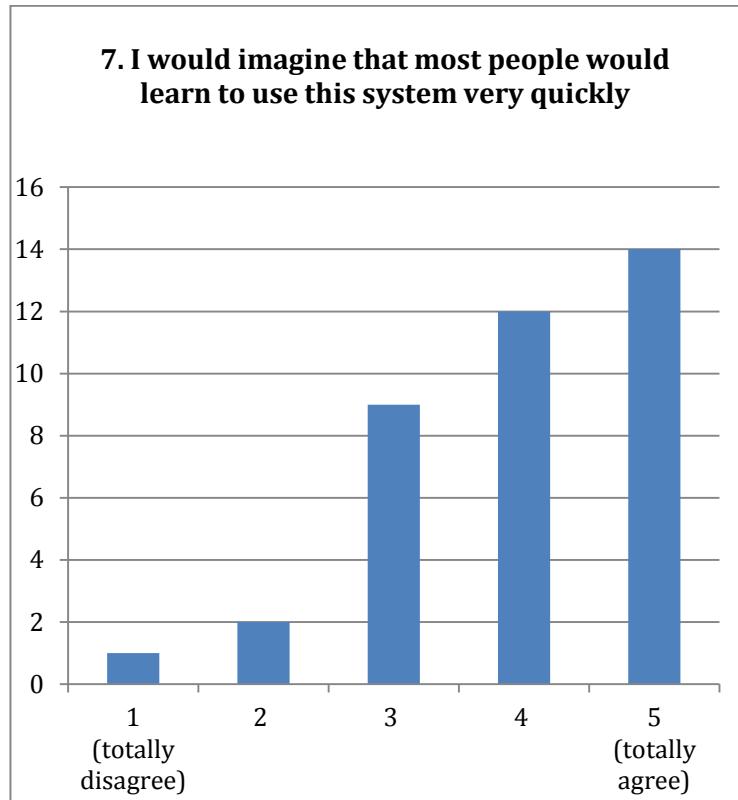
### 6.1.2. Results

The administration of the SUS questionnaires to the 38 volunteers produced the following results:









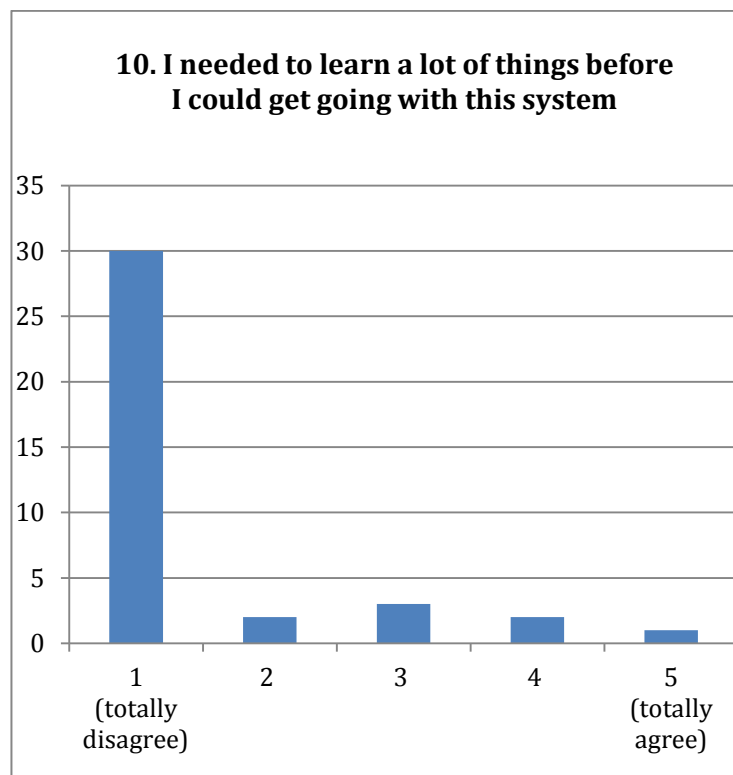
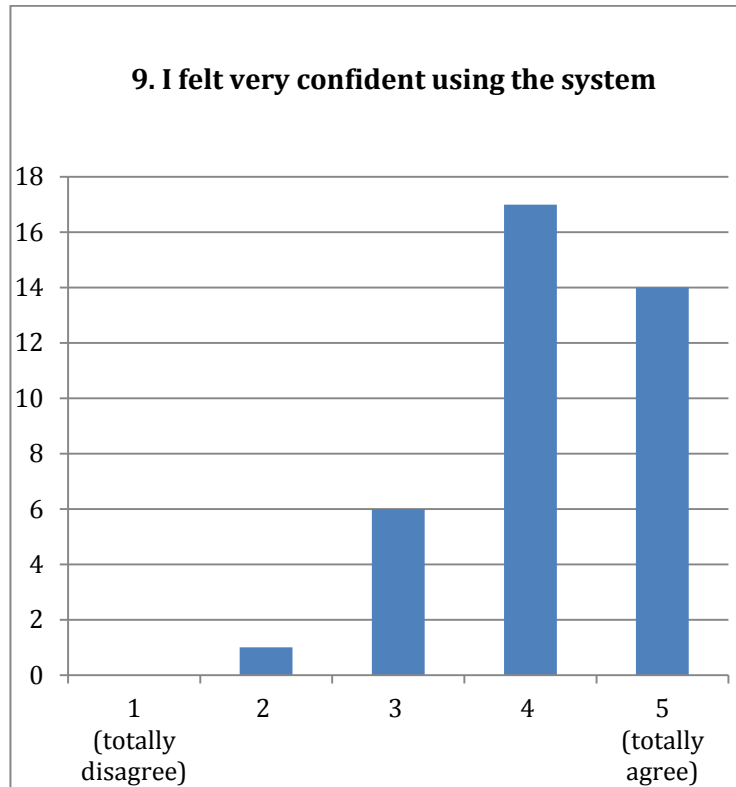


Figure 67: Results of the administration of SUS questionnaires to 38 healthy volunteers during the usability test of CoRe.

The results charted above can be summarized in the following table:

Table 5: distribution of the answers to every item of the questionnaire

	<b>ITEM</b>	<b>1 (totally disagree)</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5 (totally agree)</b>
1	I think that I would like to use this system frequently	1	2	4	15	16
2	I found the system unnecessarily complex	31	4	3	0	0
3	I thought the system was easy to use	0	3	4	12	19
4	I think that I would need the support of a technical person to be able to use this system	14	8	6	6	4
5	I found the various functions in this system were well integrated	0	0	4	10	24
6	I thought there was too much inconsistency in this system	31	5	1	1	0
7	I would imagine that most people would learn to use this system very quickly	1	2	9	12	14
8	I found the system very cumbersome to use	32	5	1	0	0
9	I felt very confident using the system	0	1	6	17	14
10	I needed to learn a lot of things before I could get going with this system	30	2	3	2	1



It can be seen in the charts and table above that the subjects were less prone to completely agree when asked about their capability and willingness to handle the system by themselves, without an expert by their side: for example, the answers for items #4, #7 and #9 are less focused on the extreme values and more spread over the middle ones, revealing a sort of "cautiousness" in the judgment. Considering other answers, according to which our system isn't complex (#2), inconsistent (#6) or cumbersome to use (#8), nor requires any previous knowledge to be used (#10), it is clear how that "cautious" feedback is not a consequence of a defect of our system, but a symptom of the already mentioned uneasiness with technology. This hypothesis is confirmed by the analysis of each single subject's profile: median answers in the "critical" items match lower levels of familiarity with a pc and/or older age; the same conclusion can also be drawn looking at the charts below (fig. 68). A table containing every subject's age and resulting SUS is available in Appendix A (A3).

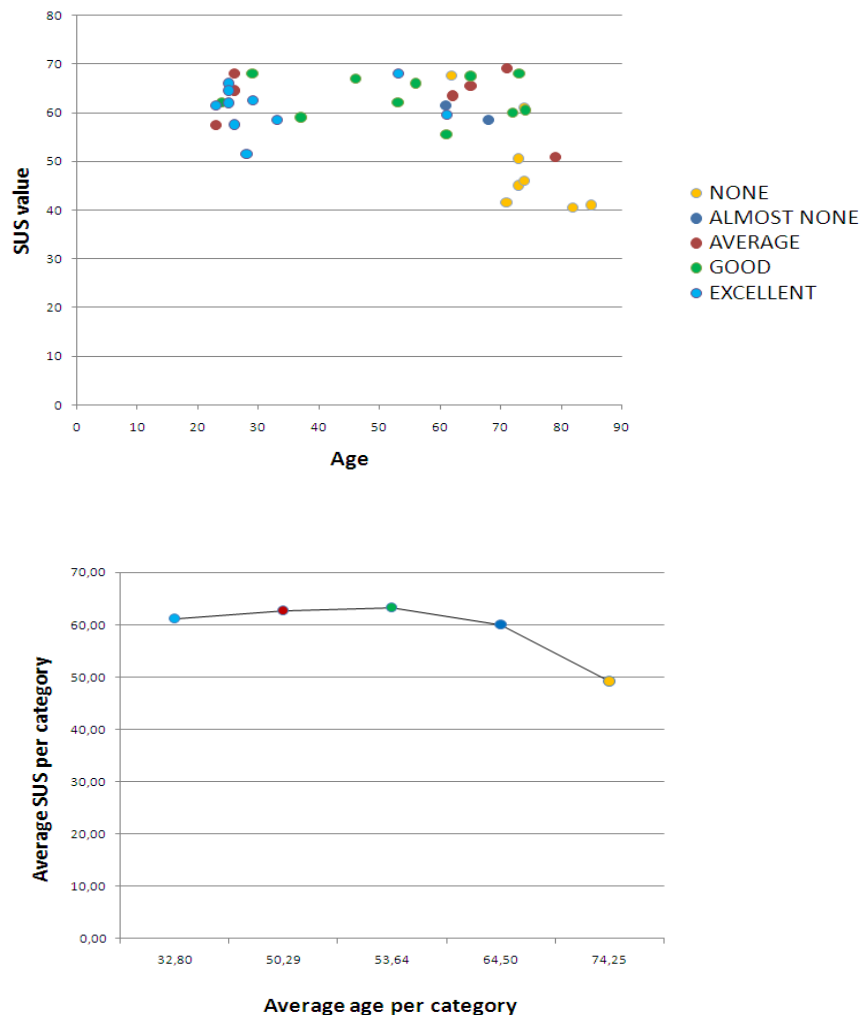


Figure 68: Chart representation of: SUS scores vs. age, above; average SUS/category vs. average age/category (of PC familiarity, represented by the color codes), below.

Subjects with excellent familiarity with a pc are the youngest ones in the group (20 to 40 years old); on the contrary, almost all the people who declared to have absolutely no familiarity with a pc are more than 70 years old. It is noticeable how the SUS score decidedly drops with the increase of age.

This realization confirmed the need of the interventions made to enhance the system, aimed at improving the involvement of the patients, in order to overcome the feeling of unease by offering a higher degree of immersion and entertainment.

## 6.2. Pilot study with a patient sample

Patients from the Mondino Foundation, afflicted by Parkinson's disease, dementia and multiple sclerosis, were the first to use CoRe throughout their rehabilitation path.

Part of the collected data, identified only by the personal id code of each subject, is presented in the following table:

Table 6: Data of the pathologic subjects enrolled for the pilot study

ID	PATHOLOGY	SIT- TINGS	DURATION (DD)	INSTAN- CES	SES- SIONS
53	PARKINSON	9	17	216	49
55	PARKINSON	12	22	367	109
56	CEREBROVASCULAR ACCIDENT	5	9	188	49
57	CEREBROVASCULAR ACCIDENT	12	26	788	128
58	DEMENTIA	4	9	240	38
59	MULTIPLE SCLEROSIS	6	16	167	31
60	PARKINSON	12	24	937	126
61	PARKINSON	12	22	797	128
62	PARKINSON	12	27	767	128
63	CEREBROVASCULAR ACCIDENT	3	2	251	38
64	PARKINSON	12	22	885	130
65	PARKINSON	12	22	916	132
66	PARKINSON	11	22	185	59
67	PARKINSON	9	21	288	69
68	CEREBROVASCULAR ACCIDENT	8	16	442	88

Of the 15 patient treated at Mondino during the enrollment period, only 9, all afflicted by Parkinson's disease, were selected to be a part of the pilot study: the other 6 were rejected either for having undergone too few sittings before discharge, or (according to the therapists) for presenting a clinical picture too different from the majority of the samples to be comparable.

Patients belonging to the sample were supposed to undergo 12 sittings in a 1-month time range. Six patients performed all the sittings, while 3 were discharged before the end of the trial and dropped out after respectively 9 and 11 sittings).

Unfortunately this reduced number of patients makes it impossible to define any statistically valuable analysis. What can be done is the observation and elaboration of the data gathered in order to derive useful information concerning the evolution of our system.

The first thing that can be noticed observing the table above is the sometimes substantial difference in the number of sessions and instances undergone, even among subjects that went through the complete set of 12 sittings. The reason for this unexpected incongruity (the tests on healthy individuals consisted in the execution of a *fixed* set of exercises) is the extreme variability of the sample: the disparity is related not only to the individual clinical pictures, but also to the temporary health conditions, which might very well negatively influence the performance and in some cases, even cause the interruption of the day's therapy.

Besides, it is natural for the therapy plan to evolve with the patients' condition: if the recovery happens, there is no reason to maintain the same plan with the same difficulty levels: this behavior would not bring any advantage and could even prove counter-productive in the long term. Table 7 shows specific performance data belonging to the 9 enrolled patients.

Table 7: Performance data of the 9 enrolled patients

ID	SIT-TINGS	ACC1	ACC0	ACC2	TOT ANSW	% (ACC 1)	% (ACC 0)	% (ACC 2)
53	9	127	86	3	216	58,80	39,81	1,39
55	12	247	120	0	367	67,30	32,70	0,00
60	12	709	228	0	937	75,67	24,33	0,00
61	12	579	218	0	797	72,65	27,35	0,00
62	12	597	170	0	767	77,84	22,16	0,00
64	12	625	260	0	885	70,62	29,38	0,00
65	12	754	162	0	916	82,31	17,69	0,00

66	11	102	81	2	185	55,14	43,78	1,08
67	9	152	136	0	288	52,78	47,22	0,00

The noticeable aspect is the predominance of correct answers (field ACC 1, that is "accuracy = 1". Acc 0 and Acc 2 represent respectively wrong and non given answers). This, per se, could be seen as a positive indicator of the quality of our interface. Had the usability been somehow lacking, there would have been a much greater number of wrong or missing answers; the great variability in the percentages is expected, considering the different conditions of the patients.

Table 8: Comparison between Response Time (sec) in patients (left) and healthy subjects (right)

<b>EXERCISE TYPE</b>	<b>avg_RT patients</b>	<b>std_dev RT patients</b>	<b>Avg RT healthy</b>	<b>std_dev RT healthy</b>
Find intruder	27,82	23,49	8,15	9,60
Word couples	89,66	63,95	19,56	10,68
Pick the element	83,94	51,90	19,20	10,60
Image and sound	5,64	6,08	3,43	3,47
Functional planning	24,98	21,88	5,46	3,49
Unscramble sentences	72,75	46,29	16,73	14,14
Unscramble images	58,46	44,39	22,43	17,46
Logical sequences	29,84	30,20	17,96	13,61
Logical analogies	22,88	16,72	6,06	4,56
Find the category	14,86	18,14	4,86	4,67

Table 8 shows not only a predictable difference between patients' and healthy volunteers' Response Times, but also offers an insight on which tasks are perceived as the most difficult: *word couples*, *pick the element* and *unscramble sentences* all take a great amount of time to be solved. Of course, being multi-click exercises, longer execution times than single-click tasks are expected. But the fact that other multi-click tasks, like *functional planning*, correspond to way shorter RT (more than 65 seconds

shorter than *word couples*) means that the cause is not to identify with the resolution type.

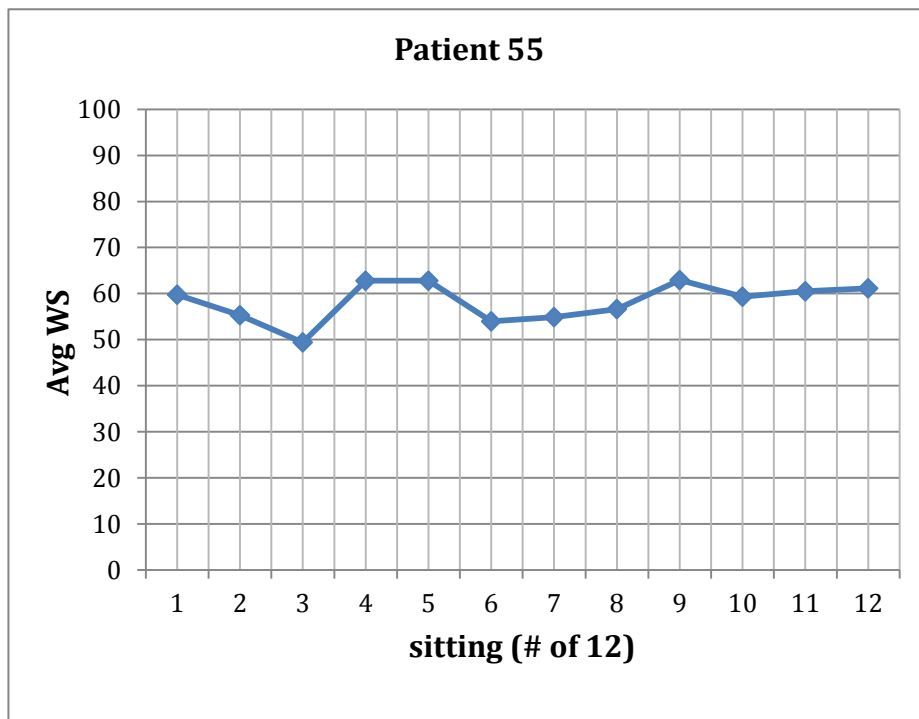
What can be observed is that the longest times are associated with exclusively textual exercises, and in particular, types with high number of stimuli visualized in every instance (8 for *word couples*, 42 for *pick the element* and 4-to-10 for *unscramble sentences*). Beside confirming the hypothesis according to which image-based tasks are more appealing and better accepted by the patients (the one that led to integration of 3D tasks into the system- see chapter 5), this verification is a clear indication of which tasks need to be re-engineered first.

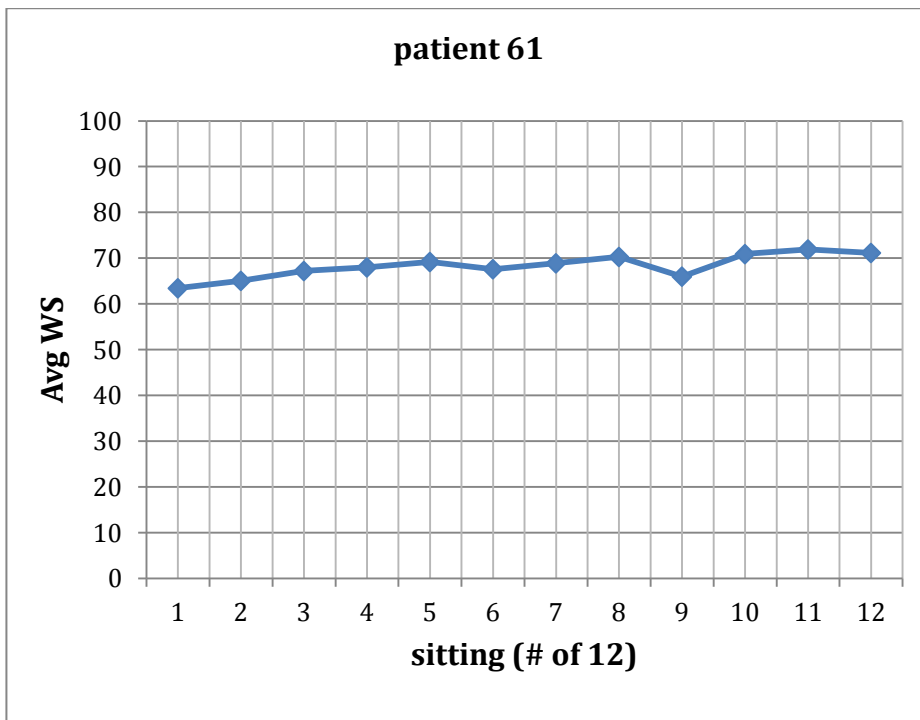
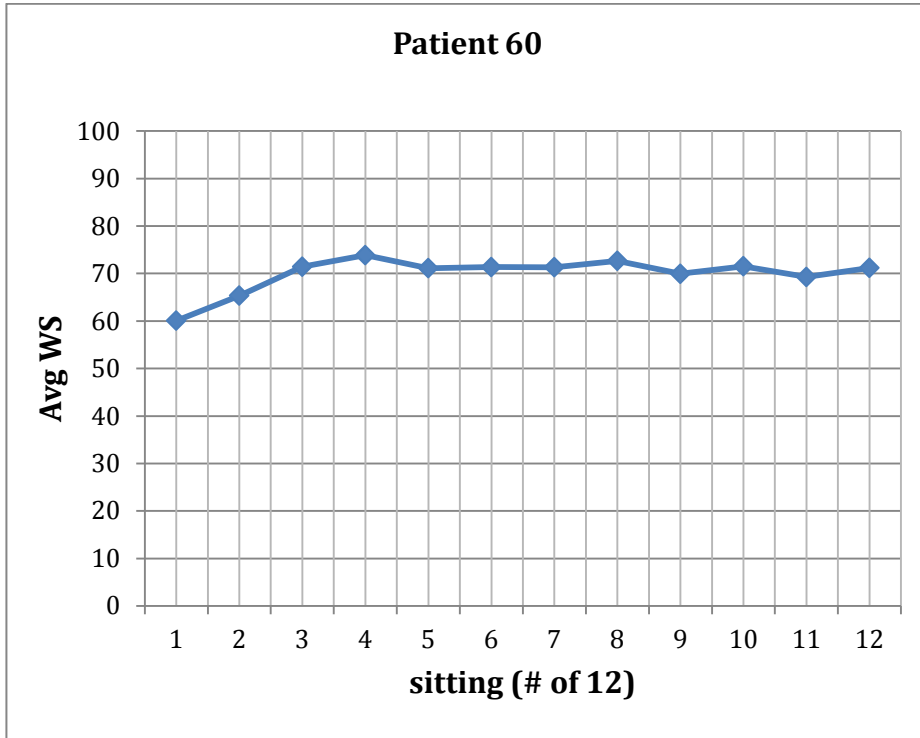
Table 9: Average and standard deviation for the number of clicks in multi-click exercises

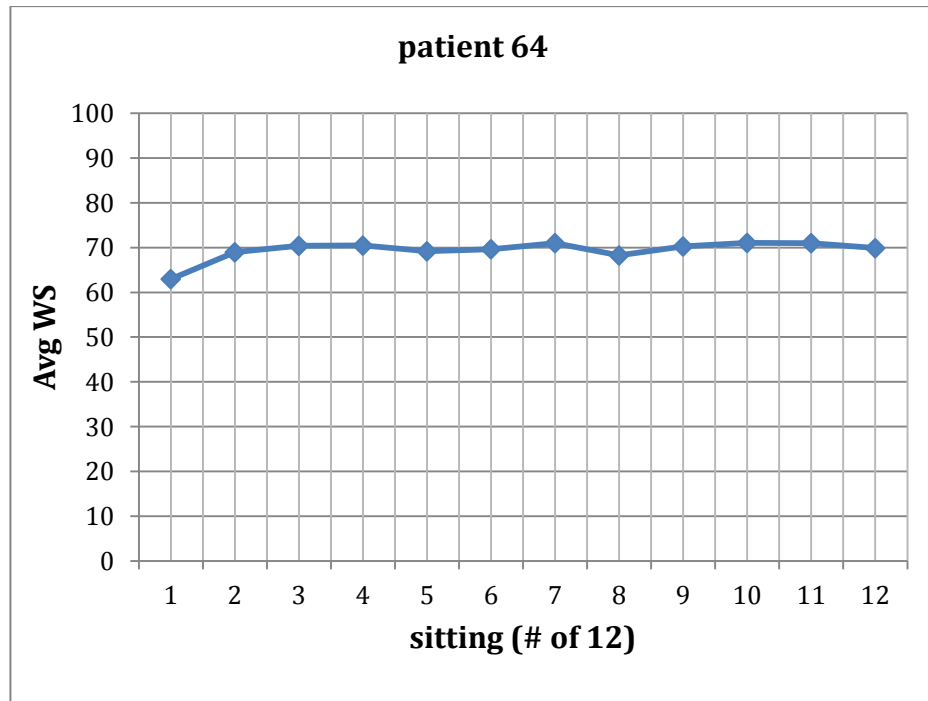
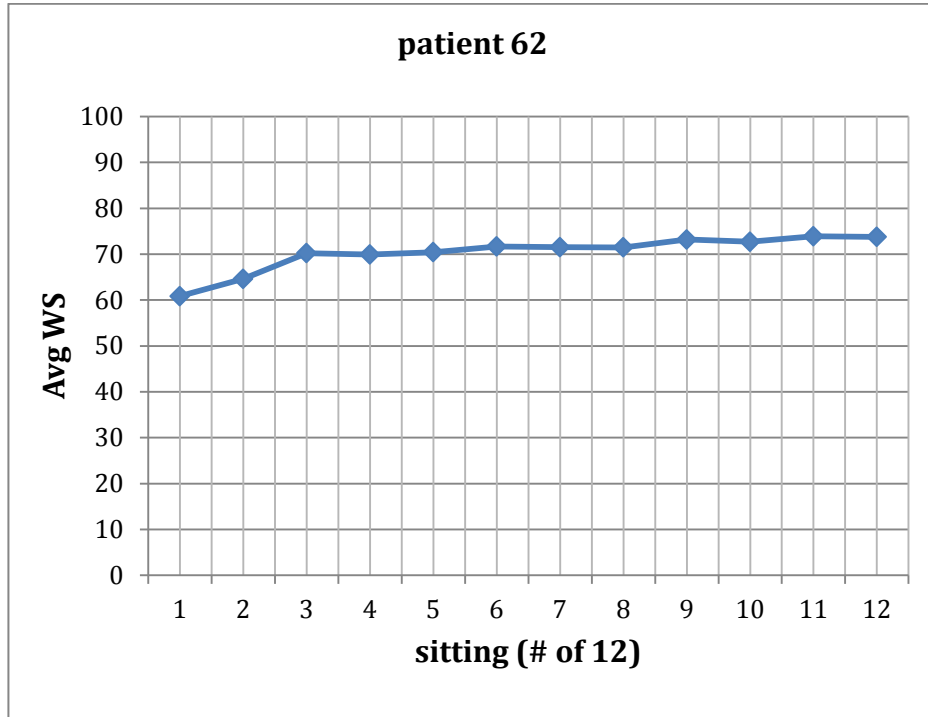
<b>EXERCISE TYPE</b>	<b>DIFF_ LEVEL</b>	<b>EXPECTED NR_CLICKS</b>	<b>AVG_NR_CLICKS</b>	<b>STD_DEV_NR_CLICKS</b>
Word_coupling	1	8	9,43	3,35
Word_coupling	2	8	9,10	2,03
Word_coupling	3	8	9,44	2,59
Pick_element	1	10	12,54	6,02
Pick_element	2	10	11,62	3,22
Pick_element	3	10	10,57	1,77
Functional_plan	1	2	2,16	0,41
Functional_plan	2	3	3,38	0,72
Functional_plan	3	2	2,19	0,49
Functional_plan	4	3	3,25	0,57
Unscramble_sentence	1	4 to 10	9,57	5,23
Unscramble_sentence	2	4 to 10	9,91	5,45
Unscramble_sentence	3	4 to 10	11,62	10,04
Unscramble_images	1	3 to 6	6,04	2,87

Table 9 shows how the average number of clicks for the multi-click exercises doesn't deviate much from the expectation (that is, the minimum number of clicks actually needed to solve the task). This can also be interpreted as a positive sign, meaning that both the implementation strategy of the exercises and the organization of the rehabilitation sessions are in general well-thought out, save minor course corrections emerged during the test stage.

Lastly, CoRe's Weighted Score (calculated with the methods explained in chapter 4) was applied to the patients' data with the aim to assess potential changes in their conditions.









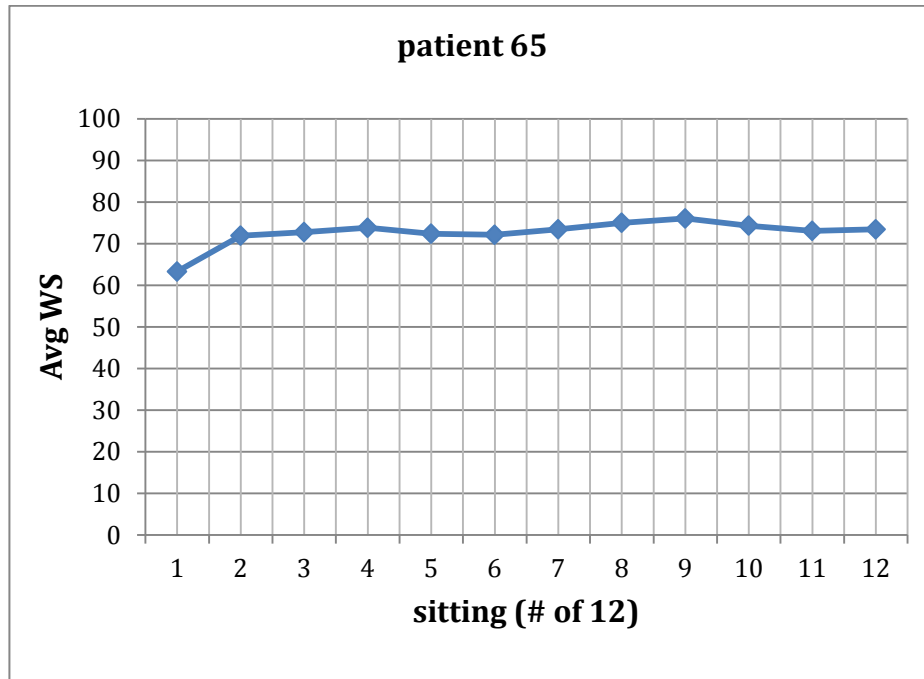


Figure 69: Charts of the average WS, created for all the patients who underwent all the 12 sittings required by the experiment.

A slight increase of the WS is noticeable in almost all the patients tested: it can be argued that the actual improvement is not sensible (around 10 points between the beginning and the conclusion of the plan), but this could be due to the many factors mentioned above, the same that discriminate patients' sittings from volunteer tests: lack of fixed protocol, differences concerning the number of instances and sessions, as well as the types of exercises and their combination and execution order, modulation of difficulty with the evolution of the patients. All of these contribute to further increase the variability of an already complex model and make any attempt to assess the tool's efficacy and (hopefully) define a standard very challenging.

# Chapter 7

---

## Conclusions and future works

### 7.1. Lesson learned

During the development of this project, some important elements emerged, which affected its progress, especially concerning the test phase and the subsequent evolutions. The most important was undoubtedly the discrepancy between the theoretical course of the development and the reality of the clinical practice.

While the test on healthy subjects went as expected, with all the volunteers undergoing all of the exercise types in all the available difficulty levels and providing an active feedback in form of questionnaire and free comments after the execution, the test on the patients was much more difficult to complete, for several reasons that will be explained below.

First of all, the availability of patients to be included in the pilot study represented the most important factual limit: only 9 cases, diagnosed with Parkinson's disease, have been included in the experimentation. And not everybody got to the end of the expected treatment (consisting in 12 sessions to be undergone in one month): 3 of them were discharged after respectively 9 and 11 sessions.

Second, it was impossible to subject the patients to the same evaluation strategy used for the test on volunteers: judgments on the usability of the system had to be derived from the observation of the patients' behavior during the sessions and the analysis of their performances in time.

Third, considering that the pilot study on patients was performed during real rehabilitation sessions, it was not possible to maintain the same activity protocol throughout the whole month of test: changes of types and difficulty levels were made following variations in the patients' conditions, which could be caused by actual improvement of their clinical picture or by temporary modifications due to external factors. Changes in drug therapy, as well as occasional fatigue or minor unrelated events could be considered causes of variation in rehabilitation protocol.

In conclusion, the scarceness of outcomes research studies and the lack of a certified clinical practice standard in the field of cognitive rehabilitation made our work all the more difficult, as well as the small number of patients enrolled for our first study made any evidence too weak to draw definitive conclusions about the effectiveness of computerized

cognitive rehabilitation of executive function. Another important factor that held back our research was the lack of homogeneity in the available pool of patients, which is a defining characteristic of our study object and thus an aspect of the matter that must be taken into account and can't be ignored.

For all these reasons it is necessary to keep this study going, hopefully extending the collaboration to other medical partners, in order to provide a sufficient number of cases to draw significant conclusions.

## **7.2. Future works: design of the clinical trial**

### **"Computer-based rehabilitation of communicative disorders in acquired cerebral injury"**

As previously mentioned, this year we started a collaboration with the IRCCS S. Maugeri Foundation to extend the use of CoRe to aphasic patients. This, as already mentioned, implies several modifications to the system currently in use before meeting the requirements of the speech therapists. Once done, it will be possible to start a clinical trial involving patients from both the Maugeri Foundation and the AO S. Gerardo (Monza). The two, together with the Laboratory of Biomedical Informatics of the University of Pavia, represent the contributors involved in a project proposal called "Computer-based rehabilitation of communicative disorders in acquired cerebral injury", submitted for acceptance by Ministero della Salute.

Below, the details of the proposal are reported.

#### **7.2.1. Overall Summary**

The project aims at improving rehabilitation of aphasic patients through the use of HomeCoRe, a computerized system that patients may learn during their hospital stay and continue using at home. HomeCore will be developed by extending CoRe, an already existing system with two distinguishing features: a big number of stimuli organized into an ontology, allowing to generate ever new exercises, so preventing patients from getting bored; automatic tailoring of exercises according to patients' characteristics, interests, and performance. New features will be the remote control of patients' compliance and performance, and the measurement of the patients' stress during exercise by opportune sensors. The efficacy of the system will be tested through a 26 months clinical trial. The two patients groups (traditional care/HomeCoRe) will be compared using a battery of standard tests, a quality of life questionnaire, and a satisfaction questionnaire, at discharge, and then every 3 months.

#### **7.2.2. Background / state of the art**

Aphasia is an acquired language impairment following brain damage, affecting some or all language modalities: expression and understanding of speech, reading and writing. About 1/3 of stroke patients (60.000 per year in Italy) experiences it, in addition to those with other types of injury (for an overall prevalence of about 200.000). There are no definite studies to support comparisons between interventions, due to difficulty in applying the methodology of the randomized controlled study, so therapists select different rehabilitation methods. However, there is evidence on the beneficial effect of linguistic cognitive therapy when it is 1) congruent with the objectives identified and 2) sufficiently prolonged/intensive. Unfortunately, due to lack of resources, access to timely and prolonged rehabilitation is difficult. To overcome these limitations, computer-based rehabilitation has been recently proposed (e.g. the tools ERICA, EVOLING, COGITO), and there is some evidence that it "...offers the potential to provide intensive home-based therapy with minimal clinician input. Improvements in performance over a number of communicative modalities can occur" [www.asha.org].

Nevertheless, not enough evidence about its advantage as homecare treatment exists until now. The importance of home rehabilitation for improving the daily living of aphasic persons fostered us to develop and evaluate this through a rigorous clinical study.

### **7.2.3. Hypothesis and Significance:**

According to the "LINEE GUIDA AFASIA FLI" (2009), evidence shows not only that patients undergoing rehabilitation have better outcomes than other subjects, but also that the treatment frequency plays a substantial role in the determination of its efficacy.

Besides, despite the need of further studies to certify their efficacy, computer-based interventions are recommended as a support to speech treatment performed in presence of a therapist.

All these statements support our hypothesis that providing a telerehabilitation system might improve clinical outcome and help overcome actual organizational issues in providing continuity of outpatient rehabilitation (difficulties in finding available therapists, reaching the hospital etc).

### **7.2.4. Specific Aims:**

- **Computerized cognitive rehabilitation (CCR) assessment-** The first aim is the assessment of feasibility and effectiveness of CCR, to be done in qualitative and quantitative terms, considering both clinical outcomes and Quality of Life (QoL) of the patients and their caregivers. This assessment task also includes evaluation of

organizational impact of the system introduction in the clinical daily routine and at patients' home.

- **Continuity of care-** A CCR aims at reinforcing the recovery process. While the speech therapist represents a pivotal element in the first stage of the treatment of communicative disorders, in further stages patients can be guided to manage their treatment more autonomously, also using new types of supports. If the CCR system is able to provide clear interface and instructions, as well as easy-to-use ergonomic devices, it is easy for the therapist to teach the patient how to use the system during the face-to-face sessions at the hospital. This way, by providing a telehomecare version of the tool, the treatment can continue after discharge: the workload of the therapist is reduced, but the exchange of information is preserved and guarantees the correct continuum of the rehabilitation.
- **Cost reduction-** CCR can improve the patient's compliance to the therapy, self confidence, autonomy, quality of life and, consequently, his medium/long-term outcome. This could result in a reduction of the Healthcare National System (NHS) costs, represented by decrease in number and frequency of control visits at the hospital, as well as a reduction of "out-of-pocket" costs (private therapists treatment).

### 7.2.5. Experimental Design

A clinical trial will be conducted to assess the CCR efficacy. Eligibility will be assessed and patients will be randomized in the "Traditional" or CCR arm. While the 1st arm patients will undergo only face-to-face encounters with speech therapists, and clinical evaluations according to traditional schedule, those in the 2nd arm will do the same, but in addition they will be trained and allowed to do more exercise by themselves using the CCR during the hospital stay, and to continue using the same tool at home.

Clinical outcomes will be evaluated by a battery of cognitive evaluation tests (see below). Quality of life will be measured by EuroQol 5L-5D questionnaire. Time needed by therapists to prepare treatment plans with the classic approach vs CCR will be compared, as well as the amount of overall rehabilitation time. According to the minimum difference, between tests scores, that is considered clinically significant, we will enroll at least 50 patients per arm in order to have a 80% statistical power and 5% significance level (e.g. to show a difference of 0.1 in utility score (SD=0.15) we need 36 cases).

All patients will undergo standard evaluation at admission, discharge and every 3 months, consisting in:

- Aachen Aphasia Test (AAT) (modified token, spontaneous language, repetition, naming, comprehension, written language test)
- Raven's Progressive Matrices Standard version (PMRS)
- Ideomotor apraxia Test

- Ideative apraxia Test
- Oral apraxia Test
- Acalculia test
- Verbal fluency test
- Other deficit-specific tests based on impairment (e.g. BADA-battery for the Analysis of Deficit in Aphasia, ENPA Neuropsychological Examination for Aphasia)
- Medical evaluation scales (FIM, GCS, BARTHEL INDEX, CRS-R)
- Assessment of type and location of the injury through CT or MR
- Quality of Life questionnaire
- Cognitive specific evaluation

Tests will be administered in their partial or complete form, based on clinical severity. Scores will be corrected for age, scholasticity and expected spontaneous improvement.

Concerning the timing, the first 8 months will be necessary to HomeCore development and installation. Since we estimate that 6-months follow-up is sufficient to measure HomeCoRe usability and efficacy, patients will then be enrolled from month 9 to month 28, in such a way to dedicate the last months to the statistical analyses.

The project will also involve a clinical trial. For patients in the CCR arm, the telemedicine system, in principle, could monitor in real time every access of the patients to their home workstations, easily measuring the frequency of use, number and total duration of completed sessions, performance, etc. However, in order to avoid boredom to patients and their caregivers, a warning message will be sent only if home treatment is not performed for an entire week. If the noncompliance persists, the therapist will personally contact the patient. Moreover, a questionnaire of "user satisfaction" will be designed and administered every 3 months, aimed to assess patient's self confidence and degree of autonomy, in order to evaluate the usability of the system.

To analyze the continuity of care also for the traditional arm patients, their compliance to the scheduled hospital control visits will be measured, and their possible use of external resources (private speech therapists or psychologists) will be investigated.

During the trial we will also monitor three types of costs in both arms:

- cost sustained by the NHS, through the number of control visits at the hospital, number and causes of re-hospitalisations, and drugs;
- "out-of-pocket" patient costs, through a quantification of the costs that patients sustain for going to control visits, such as travel expenses, costs for the caregiver, etc. Also possible costs for private therapists will be taken into account;
- Indirect costs due to productivity loss for both patients and their caregiver.

### 7.2.6. Preliminary data

Fondazione Salvatore Maugeri had a preliminary experience with CCR in the EU project MyHeart. Furthermore, the University of Pavia already implemented a CCR system (developed for other cognitive impairments) which was tested on 38 healthy volunteers at a different hospital: first the subjects underwent simulated rehabilitation sessions with the support of a therapist, and then they were administered the SUS (Software Usability Scale) questionnaire to collect their opinions about the usability of the system. The results produced useful knowledge to be exploited in this proposal.

### 7.2.7. Risk analysis, possible problems and solutions

A list of risks (r) and proposed solutions (s) is provided:

**1r.** When working autonomously, patients could interrupt rehabilitation out of boredom

**1s.** The design phase of the system will carefully consider the patients' requirements, and the system will undergo continuous improvement to meet the users' feedbacks

**2r.** In a telerehabilitation context, it is not possible to either prevent or measure external help (by caregivers) during the sessions.

**2s.** An automatic counter monitors how many times the available hints (visualization of instructions and in-game help) are provided. In addition, it is possible, with the patient's consent, to use webcam-based monitoring. This can be activated at the detection of a recent trend too positive, compared to the average performance of the patient, to ensure that no external help is provided during the execution.

**3r.** When working autonomously, the patient could start giving random answers out of boredom

**3s.** The webcam-based monitoring system (see 2.s) can be activated if events related to this scenario are detected:

- recent trend too negative for the average performance of the patient
- recent trend (latest 4 answers) of choosing the same answer
- timing out (not answering at all) the latest 2 instances
- failing at providing a feedback: when the webcam is about to be activated, a popup window appear on the screen to inform the user, requesting him to accept or refuse. If no option is selected within N seconds, the exercise ends

**4r.** The patient might suffer from slight physical impairments and not be able to use a mouse and keyboard for the execution of the exercises

4s. A touch screen is provided to him.

5r. The patient might interrupt the treatment due to worsening of clinical conditions

5s. If the interruption time is short enough (1 week max), the treatment can be resumed; if not, patient goes out of the study. Anyway, we will try to maintain the total number of enrolled patients in such a way to guarantee statistical significance of the results.

6r. The system stops working for hardware or software problems, loss of network connectivity, virus, etc.

6s. A hot-line will be provided to patients/caregivers in order to promptly solve this type of problems. Interventions at the patients' home will occur if necessary.

### **7.2.8. Expected outcomes:**

Our study is expected to highlight the usability and effectiveness of CCR system as a homecare treatment for aphasic patients.

Note that comparison of HomeCoRe versus traditional treatment will not only be based on standardized test for clinical outcomes. It is also important to evaluate, in presence of similar clinical outcomes, if the quality of life and daily living activities of patients are improved, and/or the caregivers support and direct or indirect costs are decreased. This assessment is in line with recent WHO directions that recommend using ICF (International Classification of Functioning, Disability and Health), in its widest holistic and ecological acceptance (bio-psycho-social model).

### **7.2.9. Significance and innovation**

The CCR tools developed in the past and mentioned in the "state of the art" section aimed to improve the cognitive abilities of patients with cerebral lesions, reinforcing the traditional face-to-face treatment by means of predefined exercises. Results have been encouraging, but with no definite evidence of their effectiveness in the long-term. The importance of language and related abilities stimulated us to create this system considering that the treatment is not confined to hospital stay.

Innovative aspects of HomeCoRe are:

- very personalized and ever different exercises, thanks to the ontological organization of the stimuli
- graduation of exercise difficulty according to the performance
- monitoring the home treatment after hospital discharge, in order to improve patients' motivation/compliance



Finally, with the results of this project, we could contribute to define the role of CCR in the treatment of aphasic patients, since the literature did not show clear evidence yet.

### **7.2.10. Description of the complementary and synergy research team**

The 3 operative units create an interdisciplinary and complementary team. The Neurorehabilitation unit of the Fondazione Maugeri (FSM) treats patients (mainly inpatients) with cerebral lesions including stroke. The Monza unit is a neurological rehabilitation department in a general hospital (S. Gerardo), mainly dedicated to patients discharged from the Stroke Unit (mainly outpatients)

The variety of patients with language problems is related to the severity, type, site and size of the lesion. The two hospitals will collaborate to enrol a sufficient number of patients, that will compose the two trial arms, as much as possible homogeneous.

The Biomedical Informatics Lab (BMI) of the University of Pavia collaborates since longtime with hospital teams, and in particular with FSM, where there is an internal lab of medical informatics. BMI has a great experience in designing and developing telemedicine systems and rehabilitation tools (past and present EU projects Homey, MyHeart, AP@home).

### **7.2.11. Translational relevance and impact for the National Health System (SSN)**

Investigators are increasingly aware that translational research is really a two-way street. The technological structure of the project can make innovation in the clinical practice but also the clinical practice with aphasic patients will suggest new input to improve technology-supported rehabilitation processes. This research includes different steps of translation. First is the possibility to apply the language stimuli generated by the system ontology to aphasic patients. Second is to enhance the adoption of this personalized rehabilitation treatment as best practice in the community. Collaboration of different medical and technological institutes can serve to conjugate data from clinical rehab intervention with results from studies checking efficacy/effectiveness/efficiency of technological innovations. For stakeholders, including health policy makers, systematic outcome evaluation in homecare setting can provide insight into a new model of delivering high quality rehabilitation.

## **7.3. Other future works**

### 7.3.1. HomeCoRe

Computerization of the exercises represents a step forward for the definition of personalized rehabilitation strategies, the optimization of the resources and general satisfaction of therapists and patients. A further improvement is represented by the introduction of the possibility to execute the same rehabilitation from the comfort of the patient's domicile.

This choice entails several positive aspects:

- guarantee of continuity of the rehabilitation path, following and supporting the patient's recovery not only during the hospitalization, but also after the discharge;
- possibility, if the need arises, to increase the frequency of the therapy, both in terms of number of sessions and duration of each treatment;
- ease for the patient, who can undergo the therapy when he prefers without moving from the comfortable environment that is his home;
- possibility for the therapist to handle multiple patients keeping the workload and time occupation bearable.

It must be stressed how telerehabilitation does not represent a universal solution to the problem, since not all the subjects can be considered equally eligible for this treatment. It is necessary to undergo a scrupulous neuropsychological assessment and a thorough evaluation of the patient's autonomy before deciding to introduce him to this treatment. Besides, this type of approach heavily relies on the patient's degree of compliance to the therapy: while a subject can be stimulated and convinced to focus and keep going during face to face sessions with a therapist, the same does not apply to telerehabilitation, considering that the patient is undergoing the tasks alone at his home, lacking the support and encouragement given by a human "assistant". The "human factor" of the rehabilitation is considered essential to establish a sense of self-trust in the patient, which motivates him and leads him to push on, thus establishing a virtuous circle that hopefully leads to an eventual restoration of an acceptable quality of life and a satisfactory level of independence.

Unfortunately, the very nature of the telerehabilitation- requiring that the execution of tasks happens in the absence of the specialist- denies this aspect.

For this reason, it is of the utmost importance to create a system that somehow obviates this lack.

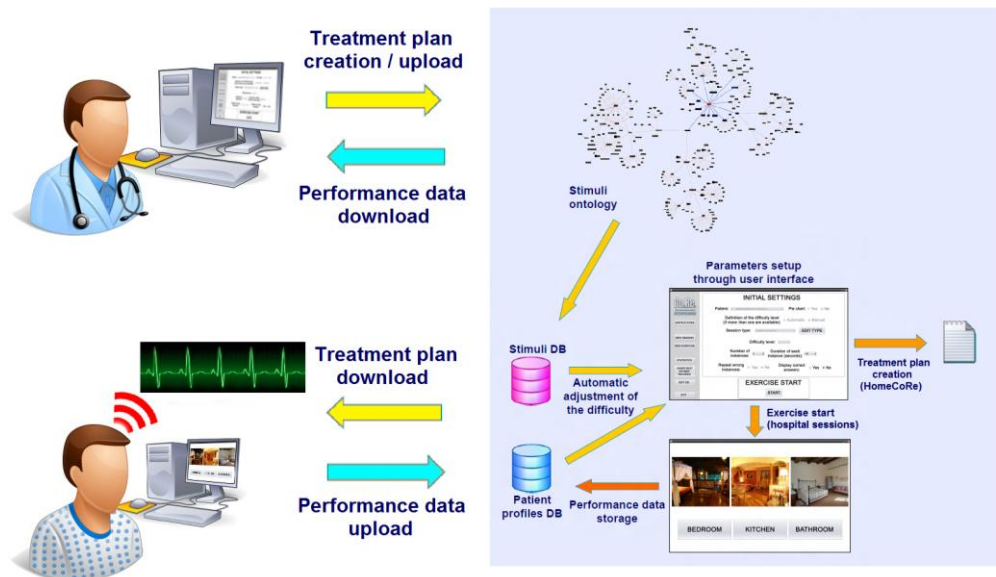
#### 7.3.1.1. Characteristics

The general structure of our system is represented in figure 70. Despite the basic system not undergoing substantial modifications, new features will be implemented, that will be integrated with the ones already existing.

First of all the possibility for the therapist to interact with the system to create treatment plans (not unlikely what he does when generating

exercises for the face-to-face hospital sessions) which must be delivered to the patient's workstation for the execution, and to download the resulting performance data once they become available.

From the patient's point of view, the same happens: it will be necessary to provide a function that allows him to download the available plans and then, after completion, to upload the resulting log files, where all the performance data and significant parameters are saved.



**Figure 70:** The structure of the HomeCoRe telerehabilitation system

Of course, it will be necessary to develop two GUIs: one for the therapist, to create the plans (it will be a slight variation of the default CoRe GUI) and one for the patient, to allow him to execute the plans.

Beside the execution parameters defined whenever a task is generated, the creation of a plan will also require to detail the temporal aspect of the therapy, that is to specify the expected total duration and frequency (sessions/days) of the treatment.

The information about the content of the plan will be saved on a file, either a simple .txt or an XML, and then sent to a server.

From the patient's point of view, whenever the system will be accessed, a scan will be performed, signaling the presence of new plans. These will be downloaded and interpreted by the software engine, which will signal the activities the patient is supposed to undergo and keep track of the performances and compliance to the proposed plan.

Of course the main requirement for the patient-side GUI is simplicity: the user must only be allowed to execute necessary operations (basically, the start of the exercises). Every other "accessory" function needs to be automated, so that no human intervention is required.

Considering that this type of functioning implies the exchange of sensitive data, the implementation of suitable security measures to protect

the transactions is essential. The creation of a VPN between the patients' and the therapists' stations might be a good solution for this issue. The application of a protocol of encryption/decryption of the information sent could be another option. The feasibility, as well as the advantages and disadvantages implied by every solution must be evaluated during the project phase of the system.

To solve the issue regarding the lack of human interaction, precautions must be taken to ensure that while executing a session the patient does not feel left alone and demotivated. Some features are going to be integrated into the system to prevent the onset of some- if not all- the unwanted consequences related to this state of mind.

Our solution consists in trying to keep the user's attention high, by implementing several types of real-time feedback from the system. This response mechanisms can only be effective when the monitoring of the patient is complete and continuous. For this purpose, the reintroduction of the stress monitoring procedure described in chapter 4 (possibly featuring real-time capabilities) could reveal extremely useful.

Besides, when feeling alone and bored, the patient could start answering randomly, just to end the session as fast as possible; or, worse, he could "cheat" and ask for the help of a caregiver. A control system must be implemented, that compares the trend of the last #n accuracy values with the average accuracy obtained from recordings of past sessions. If the result is unbalanced (a strike of #n consecutive wrong picks, a #n-times repetition of the same answer or a trend too positive if compared to the calculated average), it means something is not going well. In that case a webcam-based control system activates and records the patient to make sure the execution procedure is regular.

The same video-application can be activated whenever the maximum allowed time for execution elapses with no answer: in that case a popup window appears on screen, and a lack of feedback is considered as a strong indication that the patient is not focused on the task anymore. That's why this event triggers the automatic interruption of the session, with the specification of the reason behind the stop. The flowchart in figure 71 illustrates all the activation cases described here.

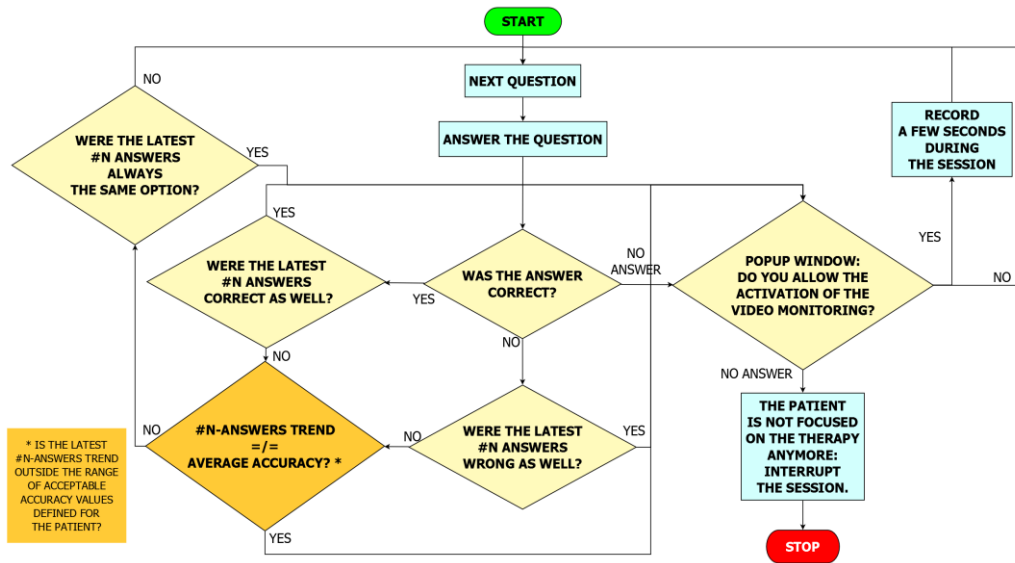


Figure 71: Flowchart of the activation of the video monitoring system

Probably the scariest event an average technology user can face when left alone to handle a software about which he knows nothing, is a system malfunctioning he is not able to handle by himself.

For this reason, a further way to support the patient and make him feel cared about is by introducing a hotline service that will provide useful advice and fast technical support in case of problems. The patient will also be able to communicate with the therapist, so that the unavoidable separation (that eventually leads to the recovery of independence in everyday life) won't be perceived as a brutal "cut", but a gradual and non traumatic detachment.

In conclusion, the long-term aim of the implementation of this cognitive rehabilitation system is the definition of statistically significant evidence that will hopefully reveal useful to favor improvements in the field of cognitive rehabilitation. Besides, if the results achieved with the telerehabilitation system match the expectations, this project will represent a step forward for the national Healthcare System, which will be affected in terms of cost reduction and resource optimization, and a general improvement of the clinical practice.

### 7.3.2. Personalization

Among the future developments of the system, the completion of the profile-based personalization (a work in progress mentioned in chapter 3) is one of the most important. After the scrupulous and systematic tagging of all the significant terms of the ontology, which is undoubtedly the most

complex and time-consuming part of the task, a suitable tag-based search algorithm will have to be created.

This solution could prove very useful for the rehabilitation process: it could become a handy tool to define the patient's level of recovery with more precision. Unlike what happens in current clinical practice (standard, generic stimuli are exploited, which may or may not have belonged to the subject's cultural baggage *even before* developing the impairment, and so potential uncertainties during the execution might be due to reasons other than the cognitive impairment), this solution does not work with a superficial, general context, but delves deeper into a specific domain, of which the patient certainly has ample knowledge. Thus, potential lacks in said knowledge can certainly be ascribed to the consequences of the neurologic pathology.

### **7.3.3. Ontology expansion**

Considering the intended future application of CoRe to language-related impairments, it will be necessary to also expand the current ontology, introducing new, grammar-related attribute fields (simple vs. compound term, word vs nonword etc.) and relations, to allow its use within exercises dedicated to aphasic patients.

### **7.3.4. Ontology editing**

To date, the only way to update the ontology is by editing the Protégé file and then loading the XML-SQL translation tool to transfer the update to the database. This solution is not particularly agile since, despite the attempt at providing a complete stimuli archive since the beginning, ontology updating is an operation that might be needed by the therapist, and the above described multi-step procedure implies either complete dependence on the developer (the one who is actually able to handle the Protégé software) or a non-negligible time investment by the therapist to learn how to use the (not particularly intuitive) Protégé User Interface. In this latter case, the risk of data loss caused by accidental erase of contents is significant.

This problem could be solved creating a new module to be integrated in the system, to bypass the need for the user to deal with Protégé. This module would be a very simplified ontology editor that would show the current structure of the schema and allow incremental updates of the leaf levels; the user would not be able to delete concepts or alter the current tree-structure in the middle layers. The update performed via this customized editor would be transferred to the ontology by appending new class-elements in the corresponding XML file. It is also possible to simplify the following step by automatically activating the XML-SQL

translator after the edit, so that the update is immediately introduced into the database as well.

### 7.3.5. CoRe3D

#### 7.3.5.1. 3D models

The advantages implied by the 3D approach to the implementation of rehabilitation exercises have already been mentioned: pseudo-natural interaction with the objects, increased entertainment and involvement of the patient into the rehabilitation process, possibility to introduce new elements into the therapy (like training of spatial perception/processing through virtual navigation, when allowed by the health condition of the patient). To achieve this, care has to be put in the modeling phase of the implementation: recognizability is fundamental, for, unlike what happens with photos, 3D models are not (and cannot be) 100% accurate representations of the original object. To date, about 50 models have been created (see figure 72), belonging to 7 different categories; the work is not over, others will be generated to enrich the archive as much as possible: this way 3D-based tasks will be allowed the same variability that text/image/sound-based exercises already show.

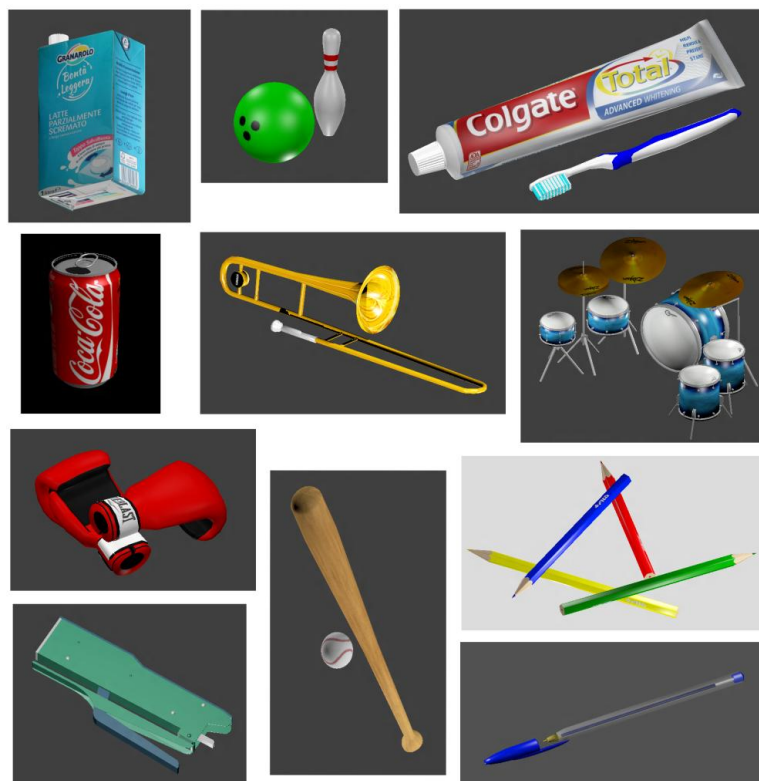


Figure 72: Some of the models created for the 3D exercises

### 7.3.5.2. Controls

To solve the still open question regarding the creation of a control system simple enough to be handled by patients, and at the same time able to provide the navigability and interactivity needed for the execution of the exercises, the introduction of particular hardware solutions must be also taken into consideration: being the manipulation of objects an essential feature for some tasks, the use of a touch screen (maybe even a multi-touch for zooming purposes) and adaptive push-button panels could represent an interesting option. As soon as a stable version of the exercises is ready to be integrated into the software system, usability tests will be undergone to assess which type of control is better accepted by the users.

### 7.3.5.3. Aids, complexity and difficulty levels

As previously stated, the walls in the "Find the intruder" game scenario are left blank by default, in order to avoid possible distractions that would interfere with the achievement of the main goal of the exercise. But if needed, an optional feature can be introduced, allowing a texture to be dynamically applied to them in order to create a meaningful context and help the patient solve the task. For example, if the main category is "kitchen supplies", the texture could show a kitchen, complete with cupboards and appliances (figure 73): this way it could be easier for the player to identify the intruder. If provided by the patient himself or his caregivers, the picture of the actual patient's kitchen could even be shown, to personalized the exercise and make the scene more familiar



Figure 73: The "Find the Intruder" task featuring the help texture



This is only one of the several aids that can be optionally introduced during the execution of an exercise (navigation ON/OFF, manipulation ON/OFF, manual actions vs. automatic animations etc.). Thus, the complexity of each session depends on many variables; for this reason the assessment of the overall difficulty level of an exercise is way more complex for 3D exercises than it is for text/picture /sound based ones and must be performed in close collaboration with the medical staff.

#### **7.3.5.4. Acceptance**

Also, it is not possible to assess which approach (text-only vs. 2D vs. 3D) will be preferred by every patient. Our goal is to offer, within one tool, the widest possible range of approaches from which to choose from. This way the therapist will be able to test different options in the early phase of the treatment and decide which one represents the most suitable solution for the specific subject.

Elderly patients (or, in general, people with little familiarity with modern technology) might refuse the 3D approach in the beginning, and switch to it later, after acquiring familiarity with the computerized therapy, while younger individuals, accustomed to playing videogames, will very likely prefer undergoing 3D exercises from the beginning.

Investigations on acceptance can be performed during the planned usability tests.



---

## Appendix A

---

Table A1: calculation of the Weighted Score based on session data. Excerpt of the table "sessioni" plus additional WS-related fields

<b>acc</b>	<b>RT</b>	<b>EXERCISE TYPE</b>	<b>Dura- tion</b>	<b>DIFF</b>	<b>ACC WS</b>	<b>DIFF WS</b>	<b>RT WS</b>	<b>TYPE WS</b>	<b>TOT WS</b>
1	9824	Find category	300	1	25	12,5	24,18133	5	66,68133
1	16845	Find category	300	1	25	12,5	23,59625	5	66,09625
1	25376	Find category	300	1	25	12,5	22,88533	5	65,38533
1	10569	Find category	300	1	25	12,5	24,11925	5	66,61925
1	23014	Find category	300	1	25	12,5	23,08217	5	65,58217
0	14097	Find category	300	1	0	12,5	23,82525	5	41,32525
1	21226	Find category	300	1	25	12,5	23,23117	5	65,73117
1	13318	Find category	300	1	25	12,5	23,89017	5	66,39017
0	130690	Functional planning	300	2	0	12,5	14,10917	5	31,60917
1	33594	Functional planning	300	2	25	12,5	22,2005	5	64,7005
1	26777	Functional planning	300	2	25	12,5	22,76858	5	65,26858
1	20747	Functional planning	300	2	25	12,5	23,27108	5	65,77108
0	34053	Functional planning	300	2	0	12,5	22,16225	5	39,66225

---

1	19262	Functional planning	300	2	25	12,5	23,39483	5	65,89483
---	-------	------------------------	-----	---	----	------	----------	---	----------

---

**AVG WS OF SESSION = 59,76552**

Table A2: Response time data per patient and exercise type (sec)

<b>ID</b>	<b>EXERCISE TYPE</b>	<b>avg_RT</b>	<b>std_dev RT</b>	<b>max_RT</b>	<b>min_RT</b>
53	Find_intruder	42,14	32,47	100,01	0,02
	Word_coupling	136,13	47,53	267,89	56,10
	Pick_element	109,38	66,81	300,00	46,58
	Functional_plan	49,54	28,32	117,16	11,77
	Unscramble_sentence	94,65	40,66	181,22	27,86
	Unscramble_images	158,75	74,68	300,00	66,48
	Logical_sequences	59,77	56,29	180,00	2,30
	Logical_analogies	47,14	36,69	186,76	12,19
	Find_category	41,74	41,29	201,87	0,01
55	Find_intruder	38,06	26,85	103,92	7,90
	Word_coupling	104,84	41,91	195,30	12,02
	Pick_element	82,74	53,77	220,15	22,47
	Image_sound	11,41	8,66	47,92	0,01
	Functional_plan	32,76	22,24	130,69	6,52
	Unscramble_sentence	110,75	49,70	240,00	23,53
	Unscramble_images	103,57	47,29	181,50	46,66
	Logical_sequences	45,63	35,18	128,15	2,09
	Logical_analogies	29,95	14,36	81,17	11,44
Find_category	18,28	13,29	58,82	0,14	
60	Find_intruder	16,21	5,61	33,19	5,82
	Word_coupling	62,30	34,47	205,39	23,62
	Pick_element	116,03	38,28	193,36	48,64
	Image_sound	3,23	2,53	27,40	1,01
	Functional_plan	17,59	10,07	47,81	5,40
	Unscramble_sentence	61,73	37,42	140,23	12,55
	Unscramble_images	45,09	28,61	108,94	11,21
	Logical_sequences	14,43	8,68	50,43	0,61
	Logical_analogies	15,30	6,58	37,25	3,75
Find_category	7,33	4,26	22,04	0,17	
61	Find_intruder	42,93	20,49	123,50	1,46

	Word_coupling	108,81	31,50	203,99	49,46
	Pick_element	64,20	27,13	126,88	19,73
	Image_sound	3,04	1,70	15,01	1,41
	Functional_plan	39,56	21,74	95,36	11,21
	Unscramble_sentence	97,72	39,88	219,64	33,52
	Unscramble_images	60,85	34,53	174,31	14,11
	Logical_sequences	19,08	13,17	84,06	4,26
	Logical_analogies	34,20	13,59	94,79	14,01
	Find_category	24,77	16,99	91,26	6,16
62	Find_intruder	24,03	12,15	62,13	3,21
	Word_coupling	63,07	35,70	203,26	20,78
	Pick_element	76,45	33,00	166,64	31,63
	Image_sound	9,29	8,60	110,33	2,84
	Functional_plan	15,44	9,48	44,56	3,74
	Unscramble_sentence	64,76	40,11	184,99	15,82
	Unscramble_images	54,76	30,37	162,59	8,50
	Logical_sequences	44,66	28,30	151,10	11,47
	Logical_analogies	28,97	13,22	85,32	11,22
	Find_category	20,32	23,78	201,44	0,41
64	Find_intruder	18,51	12,97	70,20	4,83
	Word_coupling	52,03	21,45	131,05	22,80
	Pick_element	74,65	31,41	154,80	31,77
	Image_sound	5,20	3,02	18,35	0,70
	Functional_plan	10,98	7,42	47,14	4,77
	Unscramble_sentence	62,10	40,17	216,96	14,80
	Unscramble_images	52,83	23,00	116,73	12,70
	Logical_sequences	17,56	11,75	65,00	3,10
	Logical_analogies	14,15	6,76	43,84	5,64
	Find_category	12,41	7,28	43,63	4,06
65	Find_intruder	24,32	22,62	111,24	4,52
	Word_coupling	36,24	22,77	137,93	10,70
	Pick_element	42,48	18,63	93,21	12,23
	Image_sound	4,04	3,82	31,47	1,36
	Functional_plan	12,95	15,22	90,04	3,66
	Unscramble_sentence	38,84	29,76	119,52	9,43
	Unscramble_images	34,16	30,86	156,52	3,78
	Logical_sequences	43,54	38,63	180,28	2,27
	Logical_analogies	12,33	7,52	55,27	5,28
	Find_category	5,15	4,86	37,94	1,28
66	Find_intruder	58,66	69,05	298,16	0,05
	Word_coupling	208,48	65,92	300,00	60,00
	Pick_element	130,19	71,24	300,00	47,38
	Image_sound	18,49	9,79	33,14	5,39

	Functional_plan	45,26	32,16	128,23	9,07
	Unscramble_sentence	129,28	24,01	166,31	95,68
	Unscramble_images	116,13	40,55	174,13	70,47
	Logical_sequences	51,63	31,82	137,83	13,16
	Logical_analogies	44,28	19,64	86,86	18,07
	Find_category	14,85	11,66	46,40	0,20
67	Find_intruder	39,84	17,08	103,80	18,24
	Word_coupling	182,83	67,45	300,00	91,29
	Pick_element	117,48	74,52	300,00	34,57
	Image_sound	10,10	8,91	51,52	2,13
	Functional_plan	37,33	17,18	83,06	13,80
	Unscramble_sentence	119,38	55,94	240,72	36,78
	Unscramble_images	70,88	30,48	124,24	30,30
	Logical_sequences	15,03	8,93	38,24	4,56
	Logical_analogies	24,72	11,40	62,88	9,72
	Find_category	15,64	10,48	37,37	1,51

Table A3: Resulting SUS score for every volunteer

AGE	FAMILIARITY LEVEL	SUS SCORE
26	EXCELLENT	57,5
61	GOOD	55,5
85	NONE	41
33	EXCELLENT	58,5
23	EXCELLENT	61,5
25	EXCELLENT	66
53	GOOD	62
25	EXCELLENT	64,5
26	AVERAGE	64,5
26	AVERAGE	68
62	AVERAGE	63,5

---

61	ALMOST NONE	61,5
74	GOOD	60,5
28	EXCELLENT	51,5
74	NONE	61
53	EXCELLENT	68
23	AVERAGE	57,5
29	EXCELLENT	62,5
25	EXCELLENT	62
37	GOOD	59
71	AVERAGE	69
62	NONE	67,5
61	EXCELLENT	59,5
65	AVERAGE	65,5
79	AVERAGE	51
73	NONE	50,5
82	NONE	40,5
73	NONE	45
71	NONE	41,5
74	NONE	46
72	GOOD	60
68	ALMOST NONE	58,5
46	GOOD	67
56	GOOD	66
73	GOOD	68

---

29	GOOD	68
65	GOOD	67,5
24	GOOD	62

---

---

**MEDIAN SUS  
SCORE**  
61,50

**AVERAGE SUS  
SCORE**  
59,49



---

## Appendix B

---

In this appendix the script included in the 3D-based "Find the intruder" exercise is reported. Before explaining the algorithm, some premises must be made:

- realistic relative sizes are maintained during the creation of the 3D models, so when an object must be scaled, all the others are also similarly edited to keep the common scale factor;
- the objects are not rotatable for positioning purpose: their default orientation is chosen during the creation of the models, in order to have their most recognizable side in front of the camera;
- in the blender scenario, the table is modeled as a grid of 20 x 12 squares. This interpretation helps the understanding of the algorithm, so it will be used in this explanation (the back of the table corresponds to the top row of the grid and vice versa);
- Blender allows the import of objects from external sources (other .blend files). This "append" function being callable from the python script is essential for our exercise to work correctly, because it allows the dynamic generation of the scene.

The packing algorithm implemented for the scaling and placement of objects on the table in the 3D scenario works as follows.

- The dimensions of all the objects are acquired and transformed in number of squares (rounding the result of the  $\text{obj\_dim} / \text{square\_side\_dim}$  operation to the next integer, so that partially covered squares are considered occupied), see fig. 74, bottom right;
- The total occupied area (expressed in nr. of squares) of the 5 objects is compared to the total area of the table. If it's bigger, then a scaling is performed (with scale factor =  $\text{table\_area} / \text{total\_obj\_area}$ );
- Then, for every object, its length and width are compared to the table's length and width (in fact, even if the total occupied area of the object is small, one dimension might still not fit the table); if this test fails, scaling is performed on all the objects (with scale factor =  $\text{table\_dimension} / \text{exceeding\_obj\_dimension}$ );

- Once these size tests are passed, the first object is placed, putting its back left vertex in the back left corner of the table. This way the  $X_{obj} \times Y_{obj}$  top left squares of the grid are occupied;
- The grid is scanned again, starting from the beginning, trying to position the other objects (during the search, suitable filters and restrictions are applied, depending on the size of the objects). If the  $n^{th}$  object does not find a place, the  $n-1^{th}$  element is slid forward on the grid (1 square for every cycle) and re-positioned. Then a new attempt is made to position the  $n^{th}$  object. If no configuration is found and the  $n-1^{th}$  object has been slid to the last available square with no result, the re-positioning attempt goes back one more level, sliding the  $n-2^{th}$  object forward and re-positioning the  $n-1^{th}$  and  $n^{th}$ ...

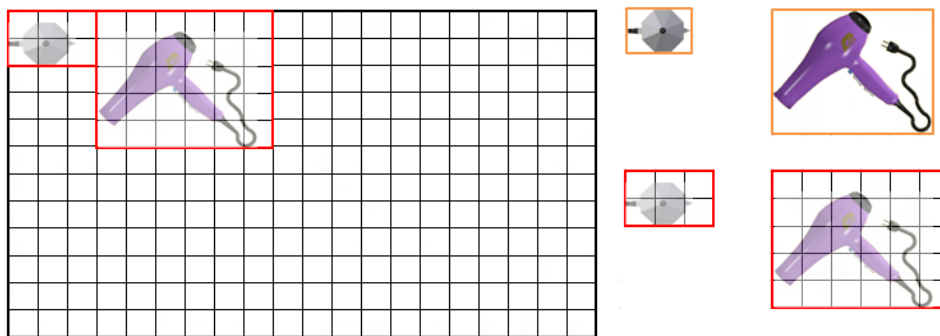


Figure 74: Representation of a step in the positioning algorithm: based on the size of their bounding box (top right), the grid occupation is defined for every object (bottom right). The positioning on the table is attempted starting from the top left corner of the grid

- If no configuration is found and the  $1^{st}$  object has been slid to the last available square, scaling is performed on all objects and the positioning cycle starts again.
- If, on the contrary, a positioning is found for all the elements, one last check is performed to verify if the objects in foreground are taller than the ones in the background: if that happens, objects in the background would be hidden during the execution of the exercise. For this reason, the positioning is restarted from scratch.

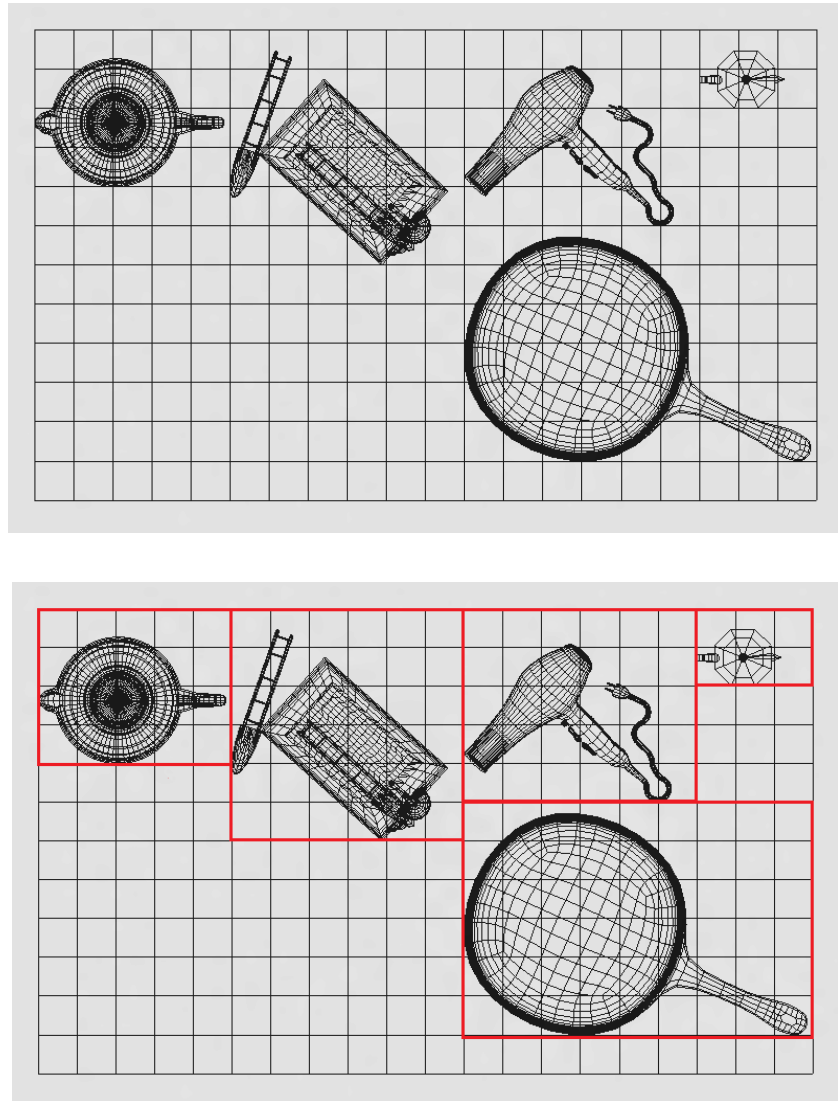


Figure 75: the final configuration of the 5 objects. Above, the pseudo-random scattering appearance; below, the actual orderly disposition based on the grid occupation of each object.

Below the whole python script is reported, including the scene setup and the algorithms for scaling and positioning:

```
import os
import bge
import math
import operator

mypath = os.path.dirname(os.path.abspath(__file__))
log = mypath + "\\blender_log.txt"
f = open(os.path.normpath(log), 'w')

os.system('cls')
```

```

# To get the controller that's running this python
script:
cont = bge.logic.getCurrentController()
# To get the game object this controller is on:
obj = cont.owner

#the py actuator is connected to an "always" controller,
so I have to be sure the procedure happens only one time
(using the "Done" variable
if obj['Done'] is True:
    print("Already TRUE!")
else:
    #obj = bge.logic.getCurrentScene().objects['Camera']
    print("False...")
    obj['Done'] = True
    print("set Done to TRUE!")

    #read a txt file having all the object to append
listed with path, object names and file name
nr_lines = 0
try:
    instance = mypath + "\\instance.txt"
    fl = open(instance, "r")
    try:
        #read one line at a time.
        #line = f.readline()
        # OR read all the lines into a list.
        next(f)
#
        lines = fl.readlines()[1:]
        nr_lines = len(lines)
    finally:
        fl.close()
except IOError:
    print("read error!")

#now I can start. For every line (object needed for the
session) I append, scale and translate on the table

#I initialize a vector that will contain the names of
the needed objects

scn = bge.logic.getCurrentScene()
spheres = ["Sphere1", "Sphere2", "Sphere3",
"Sphere4", "Sphere5"]

obj_name = []

count = 0
for line in lines:
    #first, I must get the object. A line of the
text file contains the path and objects names: I must
split that string
    print(line)
    line_split = line.split('\t')
    obj_name.append(line_split[3].rstrip('\n'))

```

```

        print("about to append: ", line_split[1], "+",
obj_name)

        print("line_split[1] = " + line_split[1] +
"\nstr(line_split[2]).strip() = " +
str(line_split[2]).strip() + "\nfilename = " +
str(line_split[3]).strip())
        #append the 5 objects I need for the session

        blend_address =
str(line_split[2]).split("\\Object")

        myobject = scn.addObject(obj_name[count],
spheres[count])

        print("-----WORLD POSITION = -----")
        print(myobject.worldPosition)
        count = count + 1

cont = bge.logic.getCurrentController()
own = cont.owner

table = scn.objects["Table"]

mesh = table.meshes[0]
print("mesh = ")
print(mesh)
# mesh = obj.getMesh("Cube") # There can be more than
one mesh...
print("vertex = ")

print(mesh.getVertexArrayLength(0))

table_vertx = []
table_verty = []

table_vert = []
table_vert_1 = []
for v_index in range(mesh.getVertexArrayLength(0)):
    table_vert.append(mesh.getVertex(0,
v_index).getXYZ())

#erase duplicates from the table_vert list
table_vert_1.append(table_vert[0])
for i in range(1,mesh.getVertexArrayLength(0)):
    for j in range(i):
        if table_vert[i] == table_vert[j]:
            break
        elif (j == i - 1):
            table_vert_1.append(table_vert[i])

print("table_vert_1 len")
print(len(table_vert_1))
table_vert = []
table_vert = table_vert_1

table_vert.sort(key = operator.itemgetter(0,1))

```

```

#define vertx, verty, vertz lists
for v_index in range(len(table_vert)):
    table_vertx.append(table_vert[v_index][0])
    table_verty.append(table_vert[v_index][1])

print("len table_vert xyz")
print(len(table_vertx))
print(len(table_verty))

table_vert_x = []
table_vert_y = []

for i in range(len(table_verty)):
    if table_vertx[i] == min(table_vertx):
        table_vert_y.append(table_verty[i])

for i in range(len(table_vertx)):
    if table_verty[i] == min(table_verty):
        table_vert_x.append(table_vertx[i])
table_vert_x.sort()
table_vert_y.sort()
table_dimensions_x = table_vert_x[len(table_vert_x)-
1] - table_vert_x[0]
table_dimensions_y = table_vert_y[len(table_vert_y)-
1] - table_vert_y[0]

##     table_dimensions_x = max(table_vertx) -
min(table_vertx)
##     table_dimensions_y = max(table_verty) -
min(table_verty)
print("table dimensions = ")
print(table_dimensions_x)
print(table_dimensions_y)

#count how many vertices per row and column the table
has
x_count = 0
y_count = 0

x_count = len(table_vert_x)
y_count = len(table_vert_y)

print("\nThe table has " + str(x_count) + " X " +
str(y_count) + " vertices on a " +
str(table_dimensions_x) + " X " +
str(table_dimensions_y) + " surface")
print(str(x_count - 1))

table_squares_x = x_count - 1
table_squares_y = y_count - 1
table_squares = table_squares_x * table_squares_y
t_square_c = table_dimensions_x / table_squares_x
t_square_r = table_dimensions_y / table_squares_y

```

---

```

    print("\nThe squares have dimensions: " +
str(table_squares_x) + " x " + str(t_square_c) + "; " +
str(table_squares_y) + " x " + str(t_square_r))

    #set a list containing the table's dimensions to
compare with the object's dimensions for resize purpose.
    #Since I don't care about the object's height as
long as it fits on the table, I only have a 2 elements
list
    table_dim = [table_dimensions_x, table_dimensions_y,
0]
    table_area = table_dim[0] * table_dim[1]

    print(table.worldTransform)
    #define the object matrix of the table to have the
vertices'position in the real world
    om_t = table.worldTransform
    t_trans = [om_t[0][3],om_t[1][3],om_t[2][3]]
    print(om_t)

    v_world_t = [[vertex[0] + t_trans[0],vertex[1] +
t_trans[1],vertex[2] + t_trans[2]] for vertex in
table_vert]

    #create a matrix containing the information about
the occupation of each square of the table grid
    occup_state= [ [ None for i in range(table_squares_x
) ] for j in range(table_squares_y) ]
    for i in range(table_squares_y):
        for j in range(table_squares_x):
            occup_state[i][j]= '.'

    obj_dim = []

    bottom_left = []

    for i in range(5):

        myObject =
bge.logic.getCurrentScene().objects[obj_name[i]]

        mesh = myObject.meshes[0]
        print("mesh = ")
        print(mesh)
        print("vertex = ")

        obj_vert = []
        obj_vert_1 = []
        obj_vertx = []
        obj_verty = []
        obj_vertz = []
        obj_vert_x = []
        obj_vert_y = []
        obj_vert_z = []

        print("nr materials")
        print(len(mesh.materials))

```

```

matID = 0

for material in mesh.materials:
#     print(mesh.getVertexArrayLength(matID))
    for v_index in
range(mesh.getVertexArrayLength(matID)):
        obj_vert.append(mesh.getVertex(matID,
v_index).getXYZ())

        matID = matID + 1

    print("*****len
obj_vert*****")
    print(len(obj_vert))
    #erase duplicates from the obj_vert list
    obj_vert_1.append(obj_vert[0])
    for i in range(1,len(obj_vert)):
        for j in range(i):
            if obj_vert[i] == obj_vert[j]:
                break
            elif (j == i - 1):
                obj_vert_1.append(obj_vert[i])

    print("*****obj_vert_1
len*****")
    print(len(obj_vert_1))

    obj_vert = []
    obj_vert = obj_vert_1

    obj_vert.sort(key = operator.itemgetter(2,1,0))

#define vertx, verty, vertz lists

for v_index in range(len(obj_vert)):
    obj_vertx.append(obj_vert[v_index][0])
    obj_verty.append(obj_vert[v_index][1])
    obj_vertz.append(obj_vert[v_index][2])

    print("len obj_vert xyz")
    print(len(obj_vertx))
    print(len(obj_verty))
    print(len(obj_vertz))

    bottom_left.append([min(obj_vertx),
min(obj_verty), min(obj_vertz)])
    print("*****bottom_left = ")
    print(bottom_left)
    for i in range(len(obj_verty)):
        if obj_vertx[i] == min(obj_vertx):
            obj_vert_y.append(obj_verty[i])

    for i in range(len(obj_vertx)):
        if obj_verty[i] == min(obj_verty):
            obj_vert_x.append(obj_vertx[i])

```



```

        for i in range(len(obj_vert)):
            if obj_vertx[i] == min(obj_vertx) and
obj_verty[i] == min(obj_verty):
                obj_vert_z.append(obj_vertz[i])

        obj_vert_x.sort()
        obj_vert_y.sort()
        obj_vert_z.sort()

        obj_dimensions_x = max(obj_vertx) -
min(obj_vertx)
        obj_dimensions_y = max(obj_verty) -
min(obj_verty)
        obj_dimensions_z = max(obj_vertz) -
min(obj_vertz)

obj_dim.append([obj_dimensions_x,obj_dimensions_y,obj_di
mensions_z])

print("obj_dim = ")
for row in obj_dim:
    print(" ".join(map(str,row)))

obj_dim_scaled = obj_dim
for i in range(5):
    print(obj_dim_scaled[i])

#number of squares
obj_squares = [ [ None for i in range(3) ] for j in
range(5) ]
tot_squares = 0

found = "N"
valid = "N"

range_c = [None,None,None,None,None]
range_r = [None,None,None,None,None]

maxattempt_c = [0,0,0,0,0]
maxattempt_r = [0,0,0,0,0]

starting_c = [-1,-1,-1,-1,-1]
starting_r = [-1,-1,-1,-1,-1]

scale = 1
rescale = [1,1,1,1,1,1,1,1,1,1]

#-----
-
def reset_all():
    global found, range_c, range_r, maxattempt_c,
maxattempt_r, starting_c, starting_r, scale, rescale
    found = "N"
    valid = "N"

    for i in range(table_squares_y):

```

```

        for j in range(table_squares_x):
            occup_state[i][j]= '.'

    range_c = [None,None,None,None,None]
    range_r = [None,None,None,None,None]

    maxattempt_c = [0,0,0,0,0]
    maxattempt_r = [0,0,0,0,0]

    starting_c = [-1,-1,-1,-1,-1]
    starting_r = [-1,-1,-1,-1,-1]

    scale = 1
    rescale = [1,1,1,1,1,1,1,1,1,1]
    return

#-----

#-----
def get_squares():
    f.write("\nEntered GET_SQUARES function")
    global obj_squares, range_c, range_r
    obj_squares_tot = 0
    for i in range(5):
        obj_squares[i][0] =
math.ceil(obj_dim_scaled[i][0]/t_square_c)
        obj_squares[i][1] =
math.ceil(obj_dim_scaled[i][1]/t_square_r)
        obj_squares[i][2] = obj_squares[i][0] *
obj_squares[i][1]

        print("new squares")
        print(obj_squares[i][0])
        print(obj_squares[i][1])
        print(obj_squares[i][2])

        f.write("\n\n New squares")
        f.write("\nObj_squares = " +
str(obj_squares[i][0]))
        f.write("\nObj_squares = " +
str(obj_squares[i][1]))

        obj_squares_tot = obj_squares_tot +
obj_squares[i][2]

        #range_c and range_r are the maximum #square
where I can place the upper left corner of the object to
fit it on the table
        range_c[i] = x_count - int(obj_squares[i][0])
        range_r[i] = y_count - int(obj_squares[i][1])

#    print("tot_squares= " + str(tot_squares))
    return obj_squares_tot
#-----

```

```

#-----
def scale_dim(scale_factor):
#   print("entered SCALE_DIM function")
    f.write("\nEntered SCALE_DIM function")

    global obj_dim_scaled

    for m in range(5):
        myObject =
bge.logic.getCurrentScene().objects[obj_name[m]]

        myObject.worldScale = scale_factor

        mesh = myObject.meshes[0]

        obj_vert = []
        obj_vertx = []
        obj_verty = []
        obj_vertz = []
        obj_vert_x = []
        obj_vert_y = []
        obj_vert_z = []

        for material in mesh.materials:
            print(mesh.getVertexArrayLength(matID))
            for v_index in
range(mesh.getVertexArrayLength(matID)):
                obj_vert.append(mesh.getVertex(matID,
v_index).getXYZ())

                    obj_vertx.append(obj_vert[v_index][0])
                    obj_verty.append(obj_vert[v_index][1])
                    obj_vertz.append(obj_vert[v_index][2])
                matID = matID + 1

            print("len(obj_vertxyz)")
            print(len(obj_vertx))
            print(len(obj_verty))
            print(len(obj_vertz))

            obj_dimensions_x = max(obj_vertx) -
min(obj_vertx)
            obj_dimensions_y = max(obj_verty) -
min(obj_verty)
            obj_dimensions_z = max(obj_vertz) -
min(obj_vertz)

            obj_dim_scaled[m] =
[obj_dimensions_x,obj_dimensions_y,obj_dimensions_z]

#         bpy.ops.transform.resize(value=(scale_factor,
scale_factor, scale_factor))
            obj_dim_scaled[m] =
[myObject.dimensions[0],myObject.dimensions[1],myObject.
dimensions[2]]

```

```

        for n in range(3):
            print(obj_dim_scaled[m])

        f.write("\n\nObj_dim_scaled = " +
str(obj_dim_scaled[m]))

#     bpy.ops.object.select_all(action='DESELECT')
    return
#-----

def rescale_all():
    f.write("\nEntered RESCALE function")
    global scale, rescale
    for objects in range(5):
        index = 2 * objects
        index1 = index + 1
        if obj_dim_scaled[objects][0] >= 1:
            if obj_dim_scaled[objects][0] -
int(obj_dim_scaled[objects][0]) >= t_square_c:
                rescale[index] =
(int(obj_dim_scaled[objects][0]) +
t_square_c)/obj_dim_scaled[objects][0]
            else:
                rescale[index] =
int(obj_dim_scaled[objects][0])/obj_dim_scaled[objects][
0]
        else:
            if obj_dim_scaled[objects][0] > t_square_c:
                rescale[index] =
t_square_c/obj_dim_scaled[objects][0]
            else:
                rescale[index] = 0

        if obj_dim_scaled[objects][1] >= 1:
            if obj_dim_scaled[objects][1] -
int(obj_dim_scaled[objects][1]) >= t_square_r:
                rescale[index1] =
(int(obj_dim_scaled[objects][1]) +
t_square_r)/obj_dim_scaled[objects][1]
            else:
                rescale[index1] =
int(obj_dim_scaled[objects][1])/obj_dim_scaled[objects][
1]
        else:
            if obj_dim_scaled[objects][1] > t_square_r:
                rescale[index1] =
t_square_r/obj_dim_scaled[objects][1]
            else:
                rescale[index1] = 0

        print(str(rescale[index]) + "\n" +
str(rescale[index1]) + "\n")
        f.write(str(rescale[index]) + "\n" +
str(rescale[index1]) + "\n")

    #I erase the rescale factor(s) = 1
    for i in range(9):

```

```

        if rescale[i] == 1:
            rescale[i] = 0
        scale = max(rescale) * max(rescale)

    print("scale factor = " + str(scale))
    f.write("\n\nscale factor = " + str(scale) + "\n\n")
    return
#-----

def position_obj(a, b, starting_obj):
    f.write("\nEntered POSITION_OBJ function")
    # print("Entered POSITION_OBJ function for object " +
    str(starting_obj))
    global found, maxattempt_r, maxattempt_c,
    occup_state, starting_r, starting_c
    f.write("\n\n*****START!
Object nr " + str(starting_obj) + "
*****\n\n")
    #free_squares is the nr of free squares which I find
    starting from a chosen position
    free_squares = 0
    #target_squares is the nr of squares I need to fit
    the object
    target_squares = int(obj_squares[starting_obj][2])
    found = "N"

    #define the starting point. each time it changes,
    the count of free squares must be restarted from 0
    for countr in range(a, range_r[starting_obj]):
        if (found == "Y"):
            break
        if countr == a + 1:
            b = 0
    # print("ENTERED THE countr LOOP - countr = " +
    str(countr))
        for countc in range(b, range_c[starting_obj]):
            if (found == "Y"):
    # print("break!")
                break
            else:
    # print("ENTERED THE countc LOOP - countc
    = " + str(countc))
                ##all of the following will have to be
    included in the two loops over a and b. For now, a = 0,
    b = 0
                free_squares = 0
                ##check if I have at least one free
    square

                maxattempt_r[starting_obj] = countr
                maxattempt_c[starting_obj] = countc

                #If the chosen starting square is
    occupied, I move to the next
                if (occup_state[countr][countc] != "."):
                    free_squares = 0
                    continue

```



```

        if found == "N":
            f.write("\nDidn't find a location for object nr
" + str(starting_obj) + "!\n")
        #     print("Didn't find a location for object nr " +
str(starting_obj) + "!")
        else:
            f.write("\nObject nr " + str(starting_obj) + "
is set!")
        #     print("\nObject nr " + str(starting_obj) + " is
set!")

    return
#-----

#-----
def unoccupy(starting_obj):
    #     print("\nEntered UNOCCUPY function for object nr "
+ str(starting_obj))
    f.write("\nEntered UNOCCUPY function")
    f.write("\n\n\nUNOCCUPY " + str(starting_obj) +
"\n\n\n")

    #reset all the squares occupied by n >= starting_obj
    for cr in range(table_squares_y):
        for cc in range(table_squares_x):
            if (not(occup_state[cr][cc] == '.')) and
(int(occup_state[cr][cc]) >= starting_obj):
                occup_state[cr][cc] = '.'

    ###     for row in occup_state:
    #         print(" ".join(map(str,row)))
    ###         f.write(" ".join(map(str,row)))
    ###         f.write("\n")

    for i in range(starting_obj + 1, 5):
        starting_r[i] = 0
        starting_c[i] = 0
    #     print("starting_r = " + str(starting_r))
    #     print("starting_c = " + str(starting_c))

    return
#-----

def evaluate_unoccupation():
    #if I'm not trying to move the first object, it's ok
    global obj_counter, start_c, start_r
    while obj_counter > 0:
        #I couldn't position the object, so I need to go
back one step and unoccupy and reposition both the
previous and the current object
        obj_counter = obj_counter - 1
        #If I already reached the maximum range allowed
for an object to move, I cannot move that object and
must try to move the previous one

```

```

        if (starting_r[obj_counter] >=
range_r[obj_counter] - 1) and (starting_c[obj_counter]
>= range_c[obj_counter] - 1):
#         print("\nCANNOT move obj nr " +
str(obj_counter) + "!!\n")
        f.write("\n\nCANNOT move obj nr " +
str(obj_counter) + "!!\n")

        continue
    else:
        unoccupy(obj_counter)

        if starting_c[obj_counter] <
range_c[obj_counter]:
            start_c = starting_c[obj_counter] + 1
            start_r = starting_r[obj_counter]
        else:
            start_c = 0
            start_r = starting_r[obj_counter] + 1

        break

    #print("start_r = " + str(start_r) + ", start_c = "
+ str(start_c))

    #but if I'm already trying to move the object nr 0,
and CANNOT, I check the ranges of all the objects
    if obj_counter == 0:
        if ((maxattempt_r[0] >= range_r[0] - 1) and
(maxattempt_c[0] >= range_c[0] - 1)) and \
            ((maxattempt_r[1] >= range_r[1] - 1) and
(maxattempt_c[1] >= range_c[1] - 1)) and \
            ((maxattempt_r[2] >= range_r[2] - 1) and
(maxattempt_c[2] >= range_c[2] - 1)) and \
            ((maxattempt_r[3] >= range_r[3] - 1) and
(maxattempt_c[3] >= range_c[3] - 1)) and \
            ((maxattempt_r[4] >= range_r[4] - 1) and
(maxattempt_c[4] >= range_c[4] - 1)):

            print("*****couldn't
set objects. Rescale*****")

f.write("\n*****couldn't set
objects. Rescale*****")

        reset_all()

        rescale_all()
        scale_dim(scale)
        get_squares()

        obj_counter = 0

        start_r = 0
        start_c = 0

```



```

    return

#-----MAIN ALGORITHM-----

tot_squares = get_squares()

#-----EVALUATE TOTAL AREA-----
-
if tot_squares > table_squares:
    f.write ("\n-----")
    ")
    f.write("\nTOTAL AREA toobig!")
    f.write ("\n-----")
    ")

    while (tot_squares > table_squares):
        scale = table_squares / tot_squares

        f.write("\ntable_squares= " +
str(table_squares))
        f.write("\nobj_squares_tot= " +
str(tot_squares))
        f.write("\nscale= " + str(scale))
#         print("table_squares= " + str(table_squares) +
", obj_squares_tot= " + str(tot_squares))
#         print("scale= " + str(scale))

        scale_dim(scale)
        tot_squares = get_squares()
        f.write("\nscale= " + str(scale))
        f.write("\nDONE!!!")
#         print("scale= " + str(scale))
#         print("DONE!!!")

f.write("\nTOTAL AREA ok!")
for i in range(5):
    f.write("\n" + str(obj_dim_scaled[i]))
#     print(obj_dim_scaled[i])

#even if the total area fits the table, if X and/or Y of
any object is bigger than the table dimensions, surely
it won't fit on the surface.
scale = 1

for i in range(5):
    scale = 1

#-----EVALUATE X-----

    while (obj_squares[i][0] > table_squares_x):
        f.write ("\n-----")
    ----")
        f.write ("\nX " + str(i) + " too big!")
        f.write ("\n-----")
    ----")

```

```

        scale = table_squares_x/obj_squares[i][0]
        f.write("\nscale= " + str(scale))

        scale_dim(scale)
        tot_squares = get_squares()
        f.write("\nscale= " + str(scale))
        f.write("\nDONE!!!")

    f.write("\nX" + str(i) + " ok!")

#-----EVALUATE Y-----
    while (obj_squares[i][1] > table_squares_y):
        f.write ("\n-----")
        f.write ("\nY " + str(i) + " too big!")
        f.write ("\n-----")

    scale = table_squares_y/obj_squares[i][1]
    f.write("\nscale= " + str(scale))

    scale_dim(scale)
    tot_squares = get_squares()
    f.write("\nscale= " + str(scale))
    f.write("\nDONE!!!")

    f.write("\nY" + str(i) + " ok!")

#-----END OF EVALUATIONS-----

#position each object on the table
obj_counter = 0

start_r = 0
start_c = 0

while valid == "N":
    print("while VALID = 'N' - obj_counter = " +
str(obj_counter))
    while obj_counter < 5:
#        print("WHILE obj_counter < 5 - obj_counter = "+
str(obj_counter))
        position_obj(start_r,start_c,obj_counter)

        #    print("\nfound = " + found + ", \nmaxattempt_r
" + str([obj_counter]) + " = " +
str(maxattempt_r[obj_counter])+ ", \nmaxattempt_c " + \
#        str([obj_counter]) + " = " +
str(maxattempt_c[obj_counter]) + ", \nstarting_r " +
str([obj_counter]) + " = " + \
#        str(starting_r[obj_counter]) + ",
\nstarting_c " + str([obj_counter]) + " = " +
str(starting_c[obj_counter]) + "\n")

        if found == "Y":
            obj_counter = obj_counter + 1

```

```

        start_r = 0
        start_c = 0
        continue
    else:
        evaluate_unoccupation()

print("found configuration!")
f.write("\nfound configuration!\n\n")
print("\n\n")

    #I found a possible configuration that fits on the
table. But now I must check the height of the objects.
    for row in range((table_squares_y - 1), 0, -1):
        if found == "N":
            break
        for column in range(table_squares_x):
            # print("column = " + str(column) +
"\nobject1 should be " + occup_state[row][column])
            # f.write("\nobject1 should be " +
occup_state[row][column])
            if (occup_state[row][column] == "."):
                # print("salto obj1")
                continue

        # print("object1 will be " +
occup_state[row][column])
        # f.write("\nobject1 will be " +
occup_state[row][column])
        object1 = int(occup_state[row][column])
        # print("object1 = " + str(object1))
        row2 = row - 1

        # print("object2 should be " +
occup_state[row2][column])
        # f.write("\nobject2 should be " +
occup_state[row2][column])
        while (occup_state[row2][column] == "."):
            # print("salto obj2")
            row2 = row2 - 1

        # print("object2 will be " +
occup_state[row2][column])
        # f.write("\nobject2 will be " +
occup_state[row2][column])
        object2 = int(occup_state[row2][column])
        # print("object2 = " + str(object2))
        if (obj_dim_scaled[object1][2] >
obj_dim_scaled[object2][2]):
            #non va bene: oggetto davanti piu alto
di quello dietro
            print("KO!!! row = " + str(row) + ",
column = " + str(column) + "\nobj1 = " + str(object1) +
" vs object2 = " + str(object2))

```

```

        f.write("KO!!! row = " + str(row) + ",
column = " + str(column) + "\nobj1 = " + str(object1) +
" vs object2 = " + str(object2))
        found = "N"
        valid = "N"
        evaluate_unoccupation()
        break

    if found == "Y":
        print("trovato la soluzione!")
        print("\n\n")
        for row in occup_state:
            f.write(" ".join(map(str,row)))
            f.write("\n")
            print (" ".join(map(str,row)))
        print("\n\n")
        valid = "Y"

#now I have the objects scaled, I have to put them on
the table-- I must define the translation vector between
the extreme (lower left) vertex of the object's bounding
box and the corresponding vertex of the table (according
to the occupation matrix)

for i in range(5):
    print("-----")
    f.write("\n-----\n")
    myObject =
bge.logic.getCurrentScene().objects[obj_name[i]]

#print (v_world_t)
v_world_t.sort(key = operator.itemgetter(1,0))

f.write("\n\nv_world_t:\n")
for row in v_world_t:
    f.write(" ".join(map(str,row)))
    f.write("\n")
#    print (" ".join(map(str,row)))
f.write("\n\n")

f.write("-.-.-.-.-")

v_world_matrix = [ [ None for i in range(x_count) ] for
j in range(y_count) ]

count_v_world_t = 0

for i in range(y_count - 1, -1, -1):
    for j in range(x_count):
        v_world_matrix[i][j] =
v_world_t[count_v_world_t]
        count_v_world_t = count_v_world_t + 1
#    print(v_world_matrix[i])
    f.write("\n\n")
    f.write("\n" + str(v_world_matrix[i]))

```

```

f.write("\n\n\nelemento = " +
str(v_world_matrix[0][0][1]))

#need to define the shift vector between the chosen
vertices
for object in range(5):
    out = 0
    myObject =
bge.logic.getCurrentScene().objects[obj_name[object]]
# myObject = bpy.data.objects[obj_name[object]]
    for i in range(y_count-1):
        #for i in range(1):
            if out == 1:
                break
            for j in range(x_count-1):
                #for j in range(1):
                    if out == 1:
                        break
                    if occup_state[i][j] == str(object):
                        for subcount_rows in range(i+1, y_count
- 1):
                            print("subcount_rows = " +
str(subcount_rows))
                            #check the next rows to see where is
the low boundary of the object
                            if (occup_state[subcount_rows][j] ==
str(object)):
                                #if the next row is also
occupied by the same object
                                if not(subcount_rows == y_count
- 2):
                                    #and it's not the last
available row
                                        continue
                                        #i keep searching
                                else:
                                    #but if it's the last row,
that square is the lower left. I take the corresponding
vertex
                                    #
                                    shift_array =
[v_world_matrix[subcount_rows+1][j][0] -
myObject.bound_box[0][0],\
#
v_world_matrix[subcount_rows+1][j][1] -
myObject.bound_box[0][1],\
#
v_world_matrix[subcount_rows+1][j][2] -
myObject.bound_box[0][2]]

print(v_world_matrix[subcount_rows+1][j][0])

print(bottom_left[object][0])
        print("\n")

print(v_world_matrix[subcount_rows+1][j][1])

print(bottom_left[object][1])

```

```

        print("\n")

print(v_world_matrix[subcount_rows+1][j][2])

print(bottom_left[object][2])
        print("\n")
        shift_array =
[v_world_matrix[subcount_rows+1][j][0] -
bottom_left[object][0],\

v_world_matrix[subcount_rows+1][j][1] -
bottom_left[object][1],\

v_world_matrix[subcount_rows+1][j][2] -
bottom_left[object][2]]
        out = 1
        break
    else:
        #I found a square in the same
row which is not occupied by the same object. So the
previous square is the lower left i need
#
        shift_array =
[v_world_matrix[subcount_rows - 1][j][0] -
myObject.bound_box[0][0],\
#
v_world_matrix[subcount_rows][j][1] -
myObject.bound_box[0][1],\
#
v_world_matrix[subcount_rows][j][2] -
myObject.bound_box[0][2]]

print(v_world_matrix[subcount_rows+1][j][0])
        print(bottom_left[object][0])
        print("\n")

print(v_world_matrix[subcount_rows+1][j][1])
        print(bottom_left[object][1])
        print("\n")

print(v_world_matrix[subcount_rows+1][j][2])
        print(bottom_left[object][2])
        print("\n")
        shift_array =
[v_world_matrix[subcount_rows - 1][j][0] -
bottom_left[object][0],\

v_world_matrix[subcount_rows][j][1] -
bottom_left[object][1],\

v_world_matrix[subcount_rows][j][2] -
bottom_left[object][2]]
        out = 1
        break

print(myObject)
print("*****shift array*****")
print(shift_array)
f.write("\n\n\nShift_array = " + str(shift_array))

```

```
myObject.worldPosition = shift_array  
print("Location:", myObject.worldPosition)
```

```
f.close()
```

---

## References

---

- [1] Emre M. “Dementia associated with Parkinson’s disease”. *Lancet Neurol* 2005; 229-237;
- [2] Manuale Merck - Morbo di Parkinson <http://www.msd-italia.it/altre/manuale/sez14/1791575.html>
- [3] Confederazione associazioni italiane Parkinson e Parkinsonismi <http://www.parkinson-italia.it>
- [4] The Michael Stern Parkinson's Research Foundation <http://www.parkinsoninfo.org/about-parkinsons-disease/>
- [5] Hughes A.J., Daniel S.E., Kilford L., Lees A.J. “Accuracy of clinical diagnosis of idiopathic Parkinson’s disease: a clinico-pathological study of 100 cases”. *J Neurol Neurosurg Psychiatry* 1992;55:181-184;
- [6] [http://www.salute.gov.it/portale/salute/p1\\_5.jsp?lingua=italiano&id=167&area=Malattie\\_del\\_sistema\\_nervoso](http://www.salute.gov.it/portale/salute/p1_5.jsp?lingua=italiano&id=167&area=Malattie_del_sistema_nervoso)
- [7] Alzheimer's Association: [http://alz.org/alzheimers\\_disease\\_1973.asp](http://alz.org/alzheimers_disease_1973.asp)
- [8] BrightFocus Foundation - Alzheimer's disease research: <http://www.brightfocus.org/alzheimers/>
- [9] S. H. Mokhtar, M. M. Bakhuraysah, D. S. Cram, and S. Petratos, “The Beta-Amyloid Protein of Alzheimer ’ s Disease : Communication Breakdown by Modifying the Neuronal Cytoskeleton,” *International Journal of Alzheimer's Disease*, vol. 2013, 2013
- [10] Ministero della Salute - dati epidemiologici malattia di Alzheimer:  
[http://www.salute.gov.it/portale/temi/p2\\_6.jsp?lingua=italiano&id=2402&area=demenze&menu=vuoto](http://www.salute.gov.it/portale/temi/p2_6.jsp?lingua=italiano&id=2402&area=demenze&menu=vuoto)
- [11] The Global Deterioration Scale for Assessment of Primary Degenerative Dementia : [www.fhca.org/members/qi/clinadmin/global.pdf](http://www.fhca.org/members/qi/clinadmin/global.pdf)



- [12] Amato M.P., Zipoli V., Portaccio E. "Multiple sclerosis-related cognitive changes: A review of cross-sectional and longitudinal studies". *J Neurol Sci* 2006; 245: 41-46;
- [13] Huijbregts S.C.J., Kalkers N.F., De Sonneville L.M.J., De Groot V., Polman C.H., "Cognitive impairment and decline in different MS subtypes". *J Neurol Sci* 2006 ; 245 :187-194;
- [14] WHO and M. S. I. Federation, "Atlas - Multiple Sclerosis resources in the world," 2008. [http://www.who.int/mental\\_health/neurology/Atlas\\_MS\\_WEB.pdf](http://www.who.int/mental_health/neurology/Atlas_MS_WEB.pdf)
- [15] National Multiple Sclerosis society: <http://www.nationalmssociety.org/>
- [16] American Speech Language Hearing Association (ASHA): <http://www.asha.org/research/>
- [17] National Aphasia Association: <http://www.aphasia.org/content/aphasia-facts>
- [18] "Aphasia Assessment," 2002. [Online]. Available: [http://www.neuropsychologycentral.com/interface/content/resources/page\\_material/resources\\_general\\_materials\\_pages/resources\\_document\\_pages/aphasia\\_assessment.pdf](http://www.neuropsychologycentral.com/interface/content/resources/page_material/resources_general_materials_pages/resources_document_pages/aphasia_assessment.pdf).
- [19] FLI - Federazione Logopedisti Italiani, "Linee guida sulla gestione logopedica del paziente afasico adulto," pp. 1–120, 2009.
- [20] G. Blanken, J. Dittmann, H. Grimm, J. C. Marshall, and C.-W. Wallesch, "Linguistic Disorders and Pathologies." 1993.
- [21] Ontario Association of Speech-Language Pathologists and Audiologists: <https://www.osla.on.ca/default.aspx?lang=1&seo=aphasia&>
- [22] Broca, P.P. (1861) Remarques sur le siège de la faculté du langage articulé, suivies d'une observation d'aphémie (perte de la parole). *Bulletins de la Société Anatomique de Paris*, 36, 330-357
- [23] BMED report " Brain Regions Can Dramatically Switch Functions": <http://www.bmedreport.com/archives/24006>

- [24] S. Canals, M. Beyerlein, H. Merkle, N.K. Logothetis. "Functional MRI evidence for LTP-induced neural network reorganization". *Curr Biol.* 2009 Mar 10;19(5):398-403. doi: 10.1016/j.cub.2009.01.037. Epub 2009 Feb 19.
- [25] S. F. Cappa, T. Benke, S. Clarke, B. Rossi, B. Stemmer, and C. M. Van Heugten, "Cognitive rehabilitation," *Eur. Handb. Neurol. Manag.*, vol. 1, pp. 545–567, 2011.
- [26] S. F. Cappa, T. Benke, S. Clarke, B. Rossi, B. Stemmer, and C. M. van Heugten, "EFNS guidelines on cognitive rehabilitation: report of an EFNS task force.," *Eur. J. Neurol.*, vol. 12, no. 9, pp. 665–80, Sep. 2005.
- [27] K. Malia, P. Law, L. Sidebottom, K. Bewick, S. Danziger, E. Schold-Davis, R. Martin-Scull, K. Murphy, and A. Vaidya, "The SCR Recommendations for Best Practice in Cognitive Rehabilitation Therapy: Acquired Brain Injury," 2004. [Online]. Available: <http://www.societyforcognitiverehab.org/membership-and-certification/documents/EditedRecsBestPrac.pdf>.
- [28] K. D. Cicerone, C. Dahlberg, K. Kalmar, D. M. Langenbahn, J. F. Malec, T. F. Bergquist, T. Felicetti, J. T. Giacino, J. P. Harley, D. E. Harrington, J. Herzog, S. Kneipp, L. Laatsch, and P. a Morse, "Evidence-based cognitive rehabilitation: recommendations for clinical practice.," *Arch. Phys. Med. Rehabil.*, vol. 81, no. 12, pp. 1596–615, Dec. 2000.
- [29] K. D. Cicerone, D. M. Langenbahn, C. Braden, J. F. Malec, K. Kalmar, M. Fraas, T. Felicetti, L. Laatsch, J. P. Harley, T. Bergquist, J. Azulay, J. Cantor, and T. Ashman, "Evidence-based cognitive rehabilitation: updated review of the literature from 2003 through 2008.," *Arch. Phys. Med. Rehabil.*, vol. 92, no. 4, pp. 519–30, Apr. 2011.
- [30] K. D. Cicerone, H. Levin, J. Malec, D. Stuss, and J. Whyte, "Cognitive rehabilitation interventions for executive function: moving from bench to bedside in patients with traumatic brain injury.," *J. Cogn. Neurosci.*, vol. 18, no. 7, pp. 1212–22, Jul. 2006.
- [31] COG.I.T.O. website: <http://cogito.integrazioni.it/cof/Index>
- [32] Brainer website: <https://www.brainer.it/>

- 
- [33] CogniPlus manual: [http://www.schuhfried.at/fileadmin/content/2\\_Kataloge\\_en/CogniPlus\\_en\\_Katalog\\_SCHUHFRIED\\_3.2.pdf](http://www.schuhfried.at/fileadmin/content/2_Kataloge_en/CogniPlus_en_Katalog_SCHUHFRIED_3.2.pdf)
- [34] CogniPlus - I programmi di Training: <http://www.schuhfried.it/cogniplus-cps/i-programmi-di-training/>
- [35] Mario Stefanelli. “Ontologie: concetti generali”. Corso di Intelligenza Artificiale in Medicina, 2009.
- [36] Mario Stefanelli. “Ontologie: linguaggi e strumenti”. Corso di Intelligenza Artificiale in Medicina, 2009.
- [37] Noy, Natalya F., and Deborah L. McGuinness. 2001. "Ontology Development 101: A Guide to Creating Your First Ontology." Stanford Knowledge Systems Laboratory Technical Report KSL-01-05 and Stanford Medical Informatics Technical Report SMI-2001-0880.
- [38] <http://www-ksl.stanford.edu/people/dlm/papers/ontology-tutorial-noymcguinness.pdf>
- [39] Classificazione Cronquist: [http://francescofiume.altervista.org/taxa/sistema\\_cronquist.pdf](http://francescofiume.altervista.org/taxa/sistema_cronquist.pdf)
- [40] W. Schneider, A. Eschman, A. Zuccolotto, “E-Prime User’s Guide”, Psychology Software Tools Inc., 2002
- [41] W. Schneider, A. Eschman, A. Zuccolotto, “E-Prime GETTING STARTED GUIDE” , Psychology Software Tools Inc., 2002
- [42] Abreu, P. F. De, Werneck, V. M. B., Costa, R. M. E. M. Da, & Carvalho, L. A. V. De. (2011). Employing Multi-agents in 3-D Game for Cognitive Stimulation. 2011 XIII Symposium on Virtual Reality, 73–78. doi:10.1109/SVR.2011.32
- [43] Grau, S., Tost, D., Campeny, R., Moya, S., & Ruiz, M. (2010). Design of 3D Virtual Neuropsychological Rehabilitation Activities. 2010 Second International Conference on Games and Virtual Worlds for Serious Applications, 109–116. doi:10.1109/VS-GAMES.2010.17
- [44] Tost, D., Grau, S., Ferré, M., García, P., Tormos, J. M., & Roig, T. (2009). PREVIRNEC : a cognitive telerehabilitation system based on Virtual Environments, 87–93.

- 
- [45] Tavanti, M., & Lind, M. (2001). 2D vs 3D, implications on spatial memory. *IEEE Symposium on Information Visualization, 2001. INFOVIS 2001.*, 2001, 139–145. doi:10.1109/INFVIS.2001.963291
- [46] Cockburn, A. (2004). Revisiting 2D vs 3D Implications on Spatial Memory, 28, 25–31.
- [47] Cockburn, A., & Mckenzie, B. (2001). 3D or not 3D? Evaluating the Effect of the Third Dimension in a Document Management System Baumgärtner, S. (2007). 2D Meets 3D: A Human-Centered Interface for Visual Data Exploration, 2273–2278.
- [48] Cockburn, A., & Mckenzie, B. (2002). Evaluating the Effectiveness of Spatial Memory in 2D and 3D Physical and Virtual Environments. In *CHI 2002* (pp. 203–210).
- [49] Dalgarno, B., & Lee, M. J. W. (2010). What are the learning affordances of 3-D virtual environments? *British Journal of Educational Technology*, 41(1), 10–32. doi:10.1111/j.1467-8535.2009.01038.x
- [50] Darken, R. P., & Durost, R. (2005). Mixed-dimension interaction in virtual environments. *Proceedings of the ACM Symposium on Virtual Reality Software and Technology - VRST '05*, 38. doi:10.1145/1101616.1101625
- [51] Baldonado, M. Q., Woodruff, A., & Kuchinsky, A. (2000). Guidelines for using multiple views in information visualization. *Proceedings of the Working Conference on Advanced Visual Interfaces - AVI '00*, 110–119. doi:10.1145/345513.345271
- [52] S. Baumgärtner, A. Ebert, M. Deller, and S. Agne, “2D Meets 3D- A Human-Centered Interface for Visual Data Exploration.” <http://dl.acm.org/citation.cfm?doid=1240866.1240993>
- [53] Bellotti, F., Berta, R., De Gloria, A., & Zappi, V. (2008). Exploring gaming mechanisms to enhance knowledge acquisition in virtual worlds. *Proceedings of the 3rd International Conference on Digital Interactive Media in Entertainment and Arts - DIMEA '08*, 77. doi:10.1145/1413634.1413653

- [54] Vidal, M., Amorim, M.-A., & Berthoz, A. (2004). Navigating in a virtual three-dimensional maze: how do egocentric and allocentric reference frames interact? *Brain Research. Cognitive Brain Research*, 19(3), 244–58. doi:10.1016/j.cogbrainres.2003.12.006
- [55] Vidal, M., Lipshits, M., McIntyre, J., & Berthoz, A. (2003). Gravity and spatial orientation in virtual 3D-mazes. *Journal of Vestibular Research : Equilibrium & Orientation*, 13(4-6), 273–86. Retrieved from <http://www.ncbi.nlm.nih.gov/pubmed/15096671>
- [56] Burigat, S., & Chittaro, L. (n.d.). Navigation in 3D Virtual Environments : Effects of User Experience and Location-pointing Navigation Aids, (August 2007).
- [57] Minocha, S., & Hardy, C. L. (2011). Designing navigation and wayfinding in 3D virtual learning spaces. *Proceedings of the 23rd Australian Computer-Human Interaction Conference on - OzCHI '11*, 211–220. doi:10.1145/2071536.2071570
- [58] Moffat, S. D., Hampson, E., & Hatzipantelis, M. (1998). Navigation in a “Virtual” Maze: Sex Differences and Correlation With Psychometric Measures of Spatial Ability in Humans, 87(519), 73–87.
- [59] Maguire, E. a, Burgess, N., & O’Keefe, J. (1999). Human spatial navigation: cognitive maps, sexual dimorphism, and neural substrates. *Current Opinion in Neurobiology*, 9(2), 171–7. Retrieved from <http://www.ncbi.nlm.nih.gov/pubmed/10322179>
- [60] Ross, S. P., Skelton, R. W., & Mueller, S. C. (2006). Gender differences in spatial navigation in virtual space: implications when using virtual environments in instruction and assessment. *Virtual Reality*, 10(3-4), 175–184. doi:10.1007/s10055-006-0041-7
- [61] Livingstone, S. a, & Skelton, R. W. (2007). Virtual environment navigation tasks and the assessment of cognitive deficits in individuals with brain injury. *Behavioural Brain Research*, 185(1), 21–31. doi:10.1016/j.bbr.2007.07.015

- [63] Blender user manual: <http://wiki.blender.org/index.php/Doc:2.6/Manual>
- [64] BlenderArtists: <http://blenderartists.org/forum/>
- [65] Sweigart, B. Al. (n.d.). *Invent Your Own Computer Games with Python*, 2<sup>nd</sup> Edition.
- [66] A. Lodi, S. Martello, and M. Monaci, “Two-dimensional packing problems: A survey,” *Eur. J. Oper. Res.*, vol. 141, no. 2, pp. 241–252, Sep. 2002.
- [67] “Enhancing computerized cognitive rehabilitation with 3D solutions”, A. Alloni, D. Tost, S. Panzarasa, C. Zucchella, S. Quaglini – Proceedings of REHAB2014, Oldenburg, 20 maggio 2014; Invited for an extended paper for the journal "Disability and Rehabilitation" Special Issue
- [68] “Applying 3D graphics to computerized cognitive rehabilitation”, A. Alloni, D. Tost, S. Panzarasa, C. Zucchella, S. Quaglini – book chapter in "Communications in Computer and Information Science", Springer (in press)

---

## List of Publications

---

- **“Patient-Tailoring of Cognitive Rehabilitation Exercises in a Homecare Setting”**, A. Alloni, S. Panzarasa, G. Leonardi, C. Zucchella, E. Sinforiani, G. Sandrini, and S. Quaglini – Proceedings of UPCP2012, Firenze, 11 agosto 2011
- **“CoRe: execution and personalization of computerized cognitive rehabilitation exercises”**, A. Alloni, S. Panzarasa, G. Sandrini, E. Sinforiani, C. Zucchella and S. Quaglini – Proceedings of GNB2012, terzo congresso del Gruppo Nazionale di Bioingegneria, Roma, 26-29 giugno 2012.
- **“Preventing boredom, fatigue and stress during computer-based cognitive rehabilitation”**, A. Alloni, S. Panzarasa, G. Sandrini, E. Sinforiani, C. Zucchella and S. Quaglini – Poster presentation at MIE2012, Pisa, 26-29 August 2012
- **“A computerized system for the rehabilitation of cognitively impaired subjects”**, A. Alloni - Presented at AIME Doctoral Consortium 2013, AIME conference Murcia, Spagna, May-June 2013
- **“Enhancing computerized cognitive rehabilitation with 3D solutions”**, A. Alloni, D. Tost, S. Panzarasa, C. Zucchella, S. Quaglini – Proceedings of REHAB2014, Oldenburg, 20 maggio 2014; *Invited for an extended paper for the journal “Disability and Rehabilitation” Special Issue*
- **“CoRe: a system for cognitive rehabilitation and its preliminary evaluation”**, A. Alloni, S. Panzarasa, C. Zucchella, E Sinforiani and S. Quaglini – Proceedings of GNB2014, quarto congresso del Gruppo Nazionale di Bioingegneria, Pavia, 25-27 giugno 2014.
- **“Applying 3D graphics to computerized cognitive rehabilitation”**, A. Alloni, D. Tost, S. Panzarasa, C. Zucchella, S. Quaglini – *book chapter in “Communications in Computer and Information Science”, Springer (in press)*