Gonzalez, J. C., Pulido, J.C., Fernández, F. A three-layer planning architecture for the autonomous control of rehabilitation therapies based on social robots. Cognitive Systems Research, 43(2017), pp. 232-249

DOI: https://doi.org/10.1016/j.cogsys.2016.09.003

A three-layer planning architecture for the autonomous control of rehabilitation therapies based on social robots ☆

José Carlos González [*,1], José Carlos Pulido [*,1], Fernando Fernández [2]

Dpto. de Informática, Universidad Carlos III de Madrid, Av. de la Universidad 30, 28911 Leganés, Madrid, Spain

Abstract

This manuscript focuses on the description of a novel cognitive architecture called NAOTherapist, which provides a social robot with enough autonomy to carry out a non contact upper limb rehabilitation therapy for patients with physical impairments, such as cerebral palsy and obstetric brachial plexus palsy. NAOTherapist comprises three levels of Automated Planning. In the high level planning, the physician establishes the parameters of the therapy such as the scheduling of the sessions, the therapeutic objectives to be achieved and certain constraints based on the medical records of the patient. This information is used to establish a customized therapy plan. The objective of the medium level planning is to execute and monitor every previous planned session with the humanoid robot. Finally, the low level planning involves the execution of path planning actions by the robot to carry out different low level instructions such as performing poses. The technical evaluation shows an accurate definition and monitoring of the therapies and sessions and a fluent interaction with the robot. This automated process is expected to save time for the professionals while guaranteeing the medical criteria.

Keywords: Robotic architecture; Human Robot Interaction; Rehabilitation therapies; Automated Planning; Socially Assistive Robotics

1. Introduction

Within the rehabilitation domain, some of the main challenges to be faced are to maintain motivation of the patients while going through long and repetitive therapies and the large amount of time required by the therapists, specially with children. The development of novel techniques and devices may be a way of addressing these challenges to ensure the progress of the patient while providing clinical support to therapeutic professionals.

The field of Socially Assistive Robotics (SAR) comprises all those robotic platforms that provide assistance to people through social interaction (Feil-Seifer & Mataric, 2005). In the area of rehabilitation, these robots have demonstrated improvements in the commitment and positive effects on the motivation of several groups of patients who suffer from physical impairments (cerebral palsy, stroke) (Fasola & Mataric, 2010; Malik, Hanapiah, Rahman, & Yussof, 2016; Tapus, Tapus, & Mataric, 2009) or cognitive disorders (autism, dementia) (Cabibihan, Javed, Ang, & Aljunied, 2013; Šabanović, Bennett, Chang, & Huber, 2013). These novel approaches are expected to obtain a better adherence to clinical

☆ This work is partially funded by grant TIN2015 65686 C5 1 R and TIN2012 38079 C03 02 of Spanish Ministerio de Economía y Competi tividad. We also are grateful to the medical team of the Hospital Virgen del Rocío in Seville (Spain) for their support and participation.

* Corresponding authors.

E mail addresses: josgonza@inf.uc3m.es (J.C. González), jcpulido@inf.uc3m.es (J.C. Pulido), ffernand@inf.uc3m.es (F. Fernández).

[1] Ph.D. Student.

[2] Lecturer.

treatment. Additionally, these systems offer novel rehabilitation tools to relieve the workload of professionals while reducing the socio-economic costs of therapy sessions. To achieve this purpose, SAR platforms should be designed taking into account some of the key objectives and challenges: the appearance of the robot, fulfilment of the clinical objectives through social interaction and the autonomy to carry out the sessions by being able to respond to unexpected situations (Tapus, Mataric, & Scasselati, 2007). Although some of the previous works attempt to fulfill these requirements, our proposal is more ambitious, since we focus on the complete autonomy of the system, always meeting the medical criteria: a clinical support tool for the automated definition of therapies adapted to each patient, together with non-teleoperated execution while monitoring the planned sessions by a social humanoid robot.

This work focuses on the field of pediatric rehabilitation for patients with motor impairment of their upper limbs, often caused by complications during pregnancy or childbirth. The weakness or loss of mobility of certain parts of the body is the consequence of a brain or nerve injury and affected individuals need rehabilitation to recover their mobility. The objective is to develop a robotic architecture that controls a humanoid robot to perform and monitor customized rehabilitation sessions based on social interaction and providing clinical measurements to professionals to evaluate the outcome of the patient. The core of the rehabilitation sessions are made up of exercises shown by the robot that the patient has to imitate. This manuscript describes the NAOTherapist robotic architecture, a SAR platform which focuses on the autonomy of the robot. It follows the general pipeline model of cognition which is heavily based on Automated Planning techniques (Ghallab, Nau, & Traverso, 2004). It has also been designed with robot-independent and domain-independent criteria.

However, the architecture is not limited to robot control, but its goal is also to support the design and monitoring of the whole robot based therapy. Therefore, our architecture also provides a clinical support tool to facilitate the therapy definition process. Before starting the rehabilitation treatment, NAOTherapist plans the schedule for the whole therapy (Pulido et al., 2014). It finds a suitable combination of exercises for each session adapted to each patient with respect to their capabilities and medical records. This ensures that each patient has a customized training that is focused on their most affected parts.

The target users of this project are children with upper-limb physical impairments caused by obstetric brachial plexus palsy, cerebral palsy or any other disorder that requires long-term rehabilitation processes. These movement disorders may threaten the quality of life and wellbeing of patients for in their daily life tasks (Dickinson et al., 2007). The majority of these patients have to live with disabilities throughout their life, and it is necessary to understand how these conditions affect each patient in order to design a personalized treatment (Krigger, 2006; Ramos & Zell, 2000). The rehabilitation program is an essential part of spasticity management (Shamsoddini, Amirsalari, Hollisaz, Rahimniya, & Khatibi-Aghda, 2014). The treatment is very hard and tiring, so developing new ways of rehabilitation for children may improve their motivation and commitment to the therapy.

NAOTherapist has been designed as a support tool for the human therapist, not a replacement. It has the assistance of the medical professionals of the Pediatric Rehabilitation Unit of the Hospital Universitario Virgen del Rocío (HUVR) in Seville (Spain). They were consulted for the definition of the therapy as well as the execution of the session. This manuscript focuses on the description and technical evaluation of the developed architecture. We want to highlight that NAOTherapist has also been tested with many healthy children in schools and it is currently involved in a long-term evaluation with a group of pediatric patients. All of these conclusions form part of near-future work.

1.1. From traditional to robotic rehabilitation procedures

In order to provide a better explanation of the target problem, it is important to understand the traditional rehabilitation procedure upon which the objectives of this work have been formulated. Fig. 1 shows the rehabilitation procedure defined in the clinical protocol of the HUVR. Each

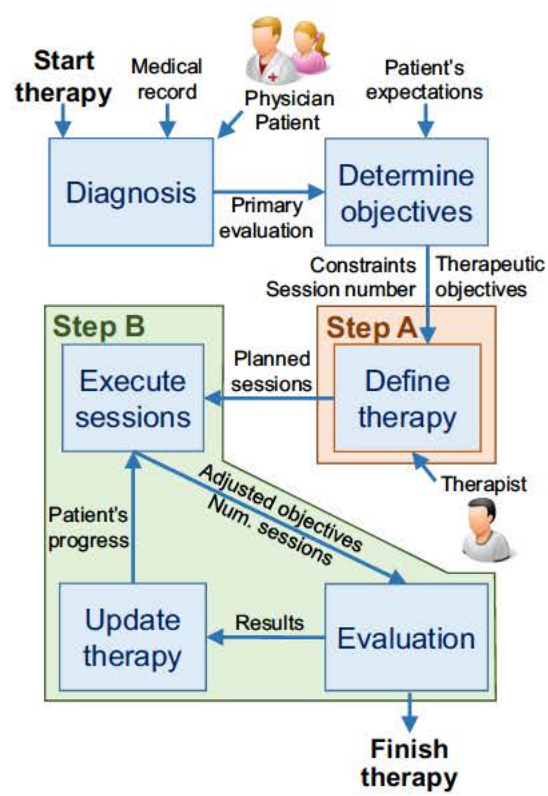

Fig. 1. The traditional rehabilitation procedure of the HUVR for patients with obstetric brachial plexus palsy and cerebral palsy.

step comprises a set of clinical guidelines that have been considered during the development of this proposal. Three main actors can be identified:

- The physician is the specialist in rehabilitation who makes the diagnosis of the patient, establishes the clinical objectives and carries out subsequent evaluations to update the therapeutic parameters if required.
- The therapist designs, guides and supervises the therapy sessions with the patient. He is in charge of guaranteeing that patients achieve their goals by encouraging them during the training.
- The patient is the primary user and beneficiary of the therapy. In this context, the patients are children with upper-limb motor disorders who have to have weekly rehabilitation sessions.

The therapeutic procedure (Fig. 1) starts with a primary evaluation of the patient according to his medical record. The results of the diagnosis together with the expectations of the patient are the elements for determining the therapeutic objectives and particular constraints of the sessions. For instance, if the patient hopes to dress or eat by himself, the physician can establish a therapy configuration suitable for the motor skills which allow the patient to achieve this goal. The progress of patients according to their expectations and desires is measured using Goal Attainment Scaling (GAS) (Turner-Stokes, 2009). This evaluation tool helps physicians to obtain a numeric estimation of the accomplishments of the patients to their specific goals. The rehabilitation procedure comprises two major steps, as can be seen in Fig. 1.

In step A, the therapeutic objectives, constraints and number of sessions are used as the input to design a full therapy plan. Planning sessions require a suitable configuration of exercises to be established that fulfills the clinical criteria. This planning step is a cumbersome task for therapists in terms of time and effort. Moreover, the design of the training plan depends greatly on each therapist and their experience. A lack of planning for the sessions may threaten the quality of the treatment and could mean that not all of the clinical aspects are covered.

Step B is the training step, in which all of the planned sessions are executed. Exercises consist of repetitive movements to strengthen the affected joints. These traditional methods may cause boredom and laziness. Therapists have to deal with this situation by investing much time and dedication getting an active engagement and commitment of the patient. Despite this effort, the development of the treatment may be tedious, so the effectiveness of the therapy is affected (Calderita et al., 2014). This situation can delay the recovery of the patient, implying a greater cost of the treatment.

The expected benefits of SAR platforms in this kind of treatment are very significant. Children perceive the robot as a friendly social entity which they can play with at the same time as they carry out their rehabilitation exercises. Active robot collaboration in these sessions is a labor-saving factor and allows the therapy supervision and monitoring process to be automated (Mataric, Eriksson, Feil-Seifer, & Winstein, 2007). They also evaluate the current motivational strategies and look for new improvements in behavior models to provide an active and fluent interaction (Nalin, Baroni, & Sanna, 2012). Furthermore, they provide an objective method for the registering of the patient in the medical records for subsequent evaluations. Other approaches based on treadmill robotic platforms provide a study of the functional effects (Drubicki et al., 2013) and demonstrate improvements after a long period of rehabilitation (Borggraefe et al., 2010).

Several challenges arise when facing the development of SAR platforms, emphasis on improving the autonomy of the robot that must react coherently to changes in the environment. A full social interaction in a dynamic environment such as a hospital requires several heterogeneous capturing devices like cameras, depth sensors, microphones or the propioception of the robot, along with different types of actuators such as motors, speech, lights or even screens. Specifically, false positives detecting complex concepts such as the intentions or emotions of the user could ruin the social experience of a SAR platform, so it is of major importance to interpret the information provided by the sensors correctly.

The objective of these requirements is not to simulate a real human social interaction perfectly, but to encourage users to believe that the robot is, in fact, a real social entity thanks to its autonomy. Normally, the behavior of these autonomous robots follows the pipeline model of cognition (perception-cognition-action), through which different approaches develop some of the basic ideas of NAOTherapist (monitoring-decision-execution) very differently.

2. Related work

The use of an architecture for Human-Robot Interaction (HRI) is a key point significant to the success of a social robot because an effective HRI platform must solve several complex problems which are very different, yet closely related. Old trends in robotics were characterized by executing low-level actions with extremely high precision, but the current research tries to perform higher level actions with acceptable results. The use of robotic frameworks such as ROS or RoboComp (Manso et al., 2010) to abstract and encapsulate multiple functionalities allows a much simpler integration of all these components and even develop cognitive architectures for robots. These architectures are the essential structure of a domain-generic computational cognitive model (Sun, 2001), so they illustrate very well the different solutions used to manage cognitive processes, in spite of the fact that these are not specifically oriented to assistance, as NAOTherapist does.

High-level knowledge has normally been represented in a symbolic way. However, there are approaches that integrate a subsymbolic version of the state of the world, which is more similar to the human cognitive experience (Avery, Kelley, & Davani, 2006; Baxter, de Greeff, & Belpaeme, 2013; Benjamin, Lyons, & Lonsdale, 2004; Trafton, Harrison, Fransen, & Bugajska, 2009). There are also specific architectures for rehabilitation which use a mixture of both representations (Prenzel, Feuser, & Gräser, 2005). The main drawback of these approaches is that subsymbolic knowledge can be difficult to be reused for other solutions, in part because only the symbolic part is directly understandable by humans. NAOTherapist uses explicitly only a symbolic representation, but it is encoded using the standard Planning Domain Definition Language (PDDL) (Fox & Long, 2003) which allows it to be used directly by any automated planner, and thus generalizing the decision support part.

Others are based on different controllers to interact with the robot (Brisben, Safos, Lockerd, Vice, & Lathan, 2005). Interestingly, some modern approaches continue to rely on simplicity and use fully reactive robotic systems without an explicit model of the state of the world (Dehkordi, Moradi, Mahmoudi, & Pouretemad, 2015). This could be useful for teleoperation or simple behaviors, but the lack of autonomy devalues one of the main challenges of SAR platforms: the capability to take decisions on the next action to be executed in a more deliberative way, without need of human intervention.

Traditional symbolic representation continues to be a significant line of research in effective SAR architectures (Boccanfuso & O'Kane, 2011; Gross et al., 2014; Mead et al., 2010; Ng-Thow-Hing, Thorisson, Sarvadevabhatla, Wormer, & List, 2009; Suárez Mejías et al., 2013). These approaches use a symbolic representation to drive rehabilitation sessions, but the deliberative part is addressed with finite-state machines. Automated Planning solutions allow increasingly complex states of the world to be managed by changing small parts in the action declaration of the PDDL domains. That eliminates the need to keep a big and coherent finite-state machine because all actions are given by the automated planner.

In a more similar approach to NAOTherapist (Galindo, Gonzalez, & Fernandez-Madrigal, 2005), the authors use a hierarchical symbolic representation of the state of the world along with a Hierarchical Task Network (HTN) planner (Nau et al., 2003). NAOTherapist includes an HTN higher level of reasoning to plan the exercises for each therapy session to be performed by the patient.

Another contribution of this work is the exploration of the independence of the robotic platform of the architecture, or more precisely all its actuators. As its name indicates, NAOTherapist has been designed for a humanoid NAO robot, but the architecture can be used in any other robot with similar characteristics just by developing a shared interface. This manuscript focuses on the application of this generic architecture to the NAO robot, but it is also been tested with other two completely different robots: the commercial platform REEM and the robot of the Ursus project (Suárez Mejías et al., 2013).

3. The NAOTherapist architecture

The architecture proposed in this work is drawn up on the basis of the clinical rehabilitation procedure explained in Fig. 1 of the previous Section 1.1, where two main steps or objectives must be achieved: the automatic definition of the therapies (Step A) and the execution of the planned sessions (Step B). To achieve these goals, we have developed a control system architecture consisting of three layers of Automated Planning that aims to execute, supervise and monitor the rehabilitation sessions with a humanoid robot (González, Pulido, Fernández, & Suárez-Mejías, 2015), while providing a clinical support tool to design therapies adapted to each patient (Pulido et al., 2014). Every execution is different from the others, since a customized problem in PDDL language is built according to the configuration of the patient. The architecture also has a knowledge base containing patient information, sessions and a catalog of exercises and postures.

NAOTherapist sessions comprise different combinations of exercises that permit those joints that are most affected to be trained specifically. The training is carried out autonomously through child robot interaction. We pursue an active engagement from the user's side, since performing these rehabilitation exercises is conditional on the will of the patient. To keep participants engaged and motivated, the robot gives them useful feedback and shows hilarious animations, and encourages them through speech throughout the training.

Section 3.1 presents a description of the three layers of planning and explains the workflow of the developed architecture. After that, the conceptual model associated with the knowledge base is detailed in Section 3.2.

3.1. The three layers of planning

According to the rehabilitation procedure in Fig. 1, step A is the process of defining or updating the therapy configuration of a certain patient and is carried out first, prior to the execution of the sessions which takes places in step B. Both tasks are addressed using Automated Planning, in which we model the problem as a set of predicates to represent the state of the world and the domain with actions or operators to change the representation of this state. Therefore, an automated planner can establish a set of actions to achieve a defined goal from an initial state. One of the advantages of using Automated Planning is that we can deal with problems with a large branching factor when there is a large number of possible actions. Moreover, the expressiveness of the models and the varied specifications of the planning languages allow each problem to be represented according to the nature of its needs; e.g. hierarchical, plain, temporal representations, etc.

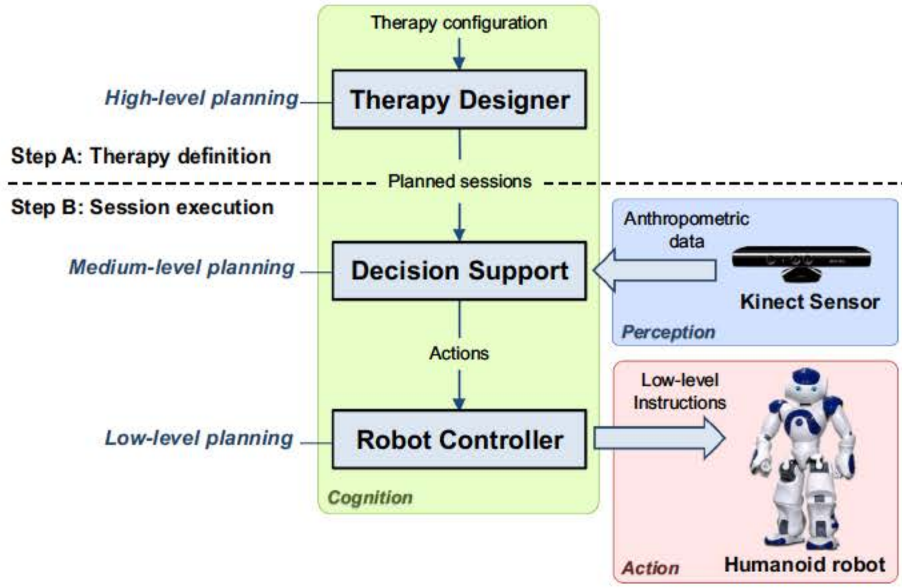

Fig. 2. The three different design criteria of the architecture: the therapeutic procedure (therapy definition and session execution steps), the pipeline model of cognition (perception, cognition and action) and the three levels of Automated Planning (high, medium and low).

The cognition process of NAOTherapist has a design based on three levels of planning, as shown in Fig. 2. A high level comprises the therapy designer that deals with the task of planning the sessions that form part of the therapy. The medium and low levels are included in the execution of the sessions. A control system is included in the medium level which is in charge of providing the necessary actions that the robot has to execute while sensing and monitoring that the received effects match the excepted ones. The perception process builds the internal state of the world. A Kinect 3D sensor serves the anthropometric data of the patient to infer information such as the correctness of the poses. In case of a mismatch between the expected and actual state of the world after the execution of a medium-level action, a replanning mechanism provides a new plan that meets the new perceived state. The low level corresponds to the path-planning mechanism to move the joints of the robot. This mechanism serves the action process and it is within the control software of the particular robot that is being used. It receives specific low-level actions that were previously transformed from the planned medium-level ones.

The NAOTherapist architecture has been created from the development of several individual components using the RoboComp robotic framework (Manso et al., 2010). The connection between components is established through TCP/IP using the Internet Communications Engine (Ice). Communications are independent of the language in which the components have been programmed because they use shared Ice interfaces. These features allow a flexible and language-independent architecture which also improves the portability of our components to other systems that pursue similar objectives.

Fig. 3 shows an overview of the architecture that plans the exercises of the whole therapy and controls individual robotic rehabilitation sessions autonomously. Each box represents a component of the architecture. On the one hand, the User Interface and Therapy Designer components correspond to the configuration and definition of the therapy. This task is addressed in the high-level planning where each session is designed according to the medical criteria of the patient. On the other hand, the Vision, Decision Support and Executive components together with the Robot and Kinect Sensor interfaces are in charge of carrying out all planned sessions while monitoring whether the execution is carried out correctly with respect to the expected effects of the planned actions. These components comprise the medium and low levels of planning, which deal with the control of the robot for the execution of each therapeutic session.

Obtaining a therapy plan is the initial step in the therapeutic procedure. In order to facilitate the configuration of therapies to the physicians, the system provides a user interface in which the expert can set up the execution parameters of the sessions. The configurable information for each therapy is mainly referred to as the schedule of the session and the objectives that are going to be trained. The therapy configuration is received by the Therapy Designer component and translated into a therapy Hierarchical Task Network (HTN) problem. This is a search an d selection task in which exercises are going to be included in the session for the specific patient while preserving the variability and session constrains (Pulido et al., 2014). Variability is important because it reduces boredom and allows the desired poses to be achieved and to train coordination in different ways. The selected exercises must be distributed throughout the session according to their intensity

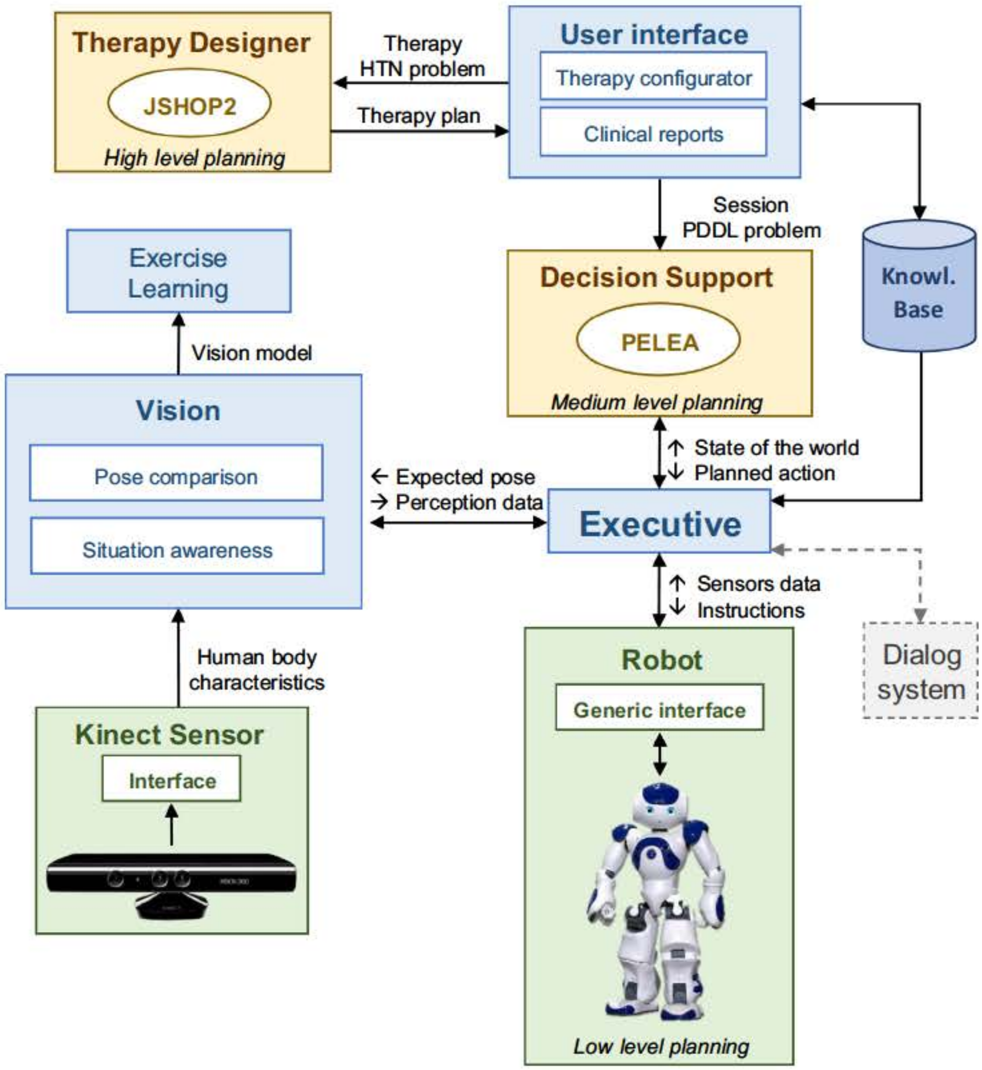

Fig. 3. NAOTherapist architecture overview.

and difficulty, so that a session is divided into three phases: warming up, training and cooling down. The planning process is carried out by the JSHOP2 HTN automated planner (Nau et al., 2003) to obtain a valid plan with the exercises for each session. A hierarchical model represents the selection process of exercises. This component behaves like a Clinical Decision Support System (CDDS) to provide assistance in the therapy design. If there are no exercises available, this module is able to suggest new exercises whose attributes comply with the established requirements and medical criteria. The high-level planning task is explained in detail in Section 4.

Once the therapy is designed, the system stores all sessions associated to the patient and the user interface allows the physician to select the next one to be executed. Each session configuration is previously translated into a planning problem associated to a classical planning domain in PDDL (Planning Domain Definition Language) (Fox & Long, 2003). The domain is modeled with respect to the objectives and requirements that the robot has to accomplish during the therapeutic session. So, it considers the set of actions that the robot can perform in each session and possible unexpected situations. The Decision Support component uses this domain and receives the problem with the session configuration to provide a valid plan of actions that meets the objectives of the training. This component is controlled by the PELEA architecture (Alcázar et al., 2010) which is in charge of planning and monitoring the execution of exercises and, if required, making decisions with respect to an unexpected perceived state. Before executing each subsequent action, PELEA compares the received state with the internal expected state. The Vision component serves the Executive component perception data about the pose and the situation of the patient in order to build the perceived state. Each action provided by Decision Support component is broken down into low level instructions by the Executive component that are sent to the Robot interface component. For instance, moving the arms to a certain pose, changing the eye color, executing animation, etc. The RoboComp paradigm allows

NAOTherapist to be independent of the robotic platform because the communication with the robot is carried out through a generic interface. This allows similar humanoid robots to work with the architecture. The use case of NAOTherapist and the session planning and execution mechanisms are detailed in Section 5.

Currently, the user interface has been improved with a monitoring module which shows session information in real time, and it is also equipped with a report tool that collects the session data of the patient and builds clinical reports based on the Quality of Upper Extremity Skills Test (QUEST) (Dematteo & Pollock, 1992; Martín et al., 2015). The Dialog component is still in progress and it may enrich the quality of the speech and interaction, since the architecture is not currently able to maintain a conversation with the users.

3.2. Conceptual model of the knowledge base

The conceptual model of NAOTherapist, shown in Fig. 4, is designed according to the project requirements proposed by the clinical experts of HUVR. The ontology tries to join all the clinical concepts with the interaction elements in order to provide a model that contains both parts meeting the project criteria. The conceptual model represents the information that is contained in the knowledge base and it is used by the architecture both in the therapy definition and in the session execution. All exercises and poses of NAOTherapist are designed by physicians and their attributes are defined according to the nature of the exercise and the subjective experience of clinical experts when children are carrying out the sessions. The stored data is crucial for the definition of therapies and the execution of the sessions.

Three roles meet in the development of a physical therapy: patient, physician and robot. Following the conceptual model in Fig. 4, a patient performs a physical therapy that is driven by a robot while is supervised by a physician. These three classes have a unique identification to distinguish the different instances in the knowledge base. The Patient class considers other useful information, both personal and clinical data, such as the comparison threshold that refers to the value that is used as the baseline to compare the poses of the exercises carried out. A physical therapy comprises a number of sessions that take place weekly at the hospital, and each session consists of a group of exercises adapted to each patient. Exercises are modeled as a sequence of poses with a specific duration and the posture associated to both arms. This represents the decomposition from a physical therapy to the order of postures through which exercises of sessions are made up. A posture is defined by a set of joint angles and other attributes, such as speech or description, which improve the interaction with the user. A speech attribute is also considered in the exercise class which is useful in clarifying clearer what the users have to do.

In order to have a more accurate model for the definition of the therapy, the ontology considers which domains

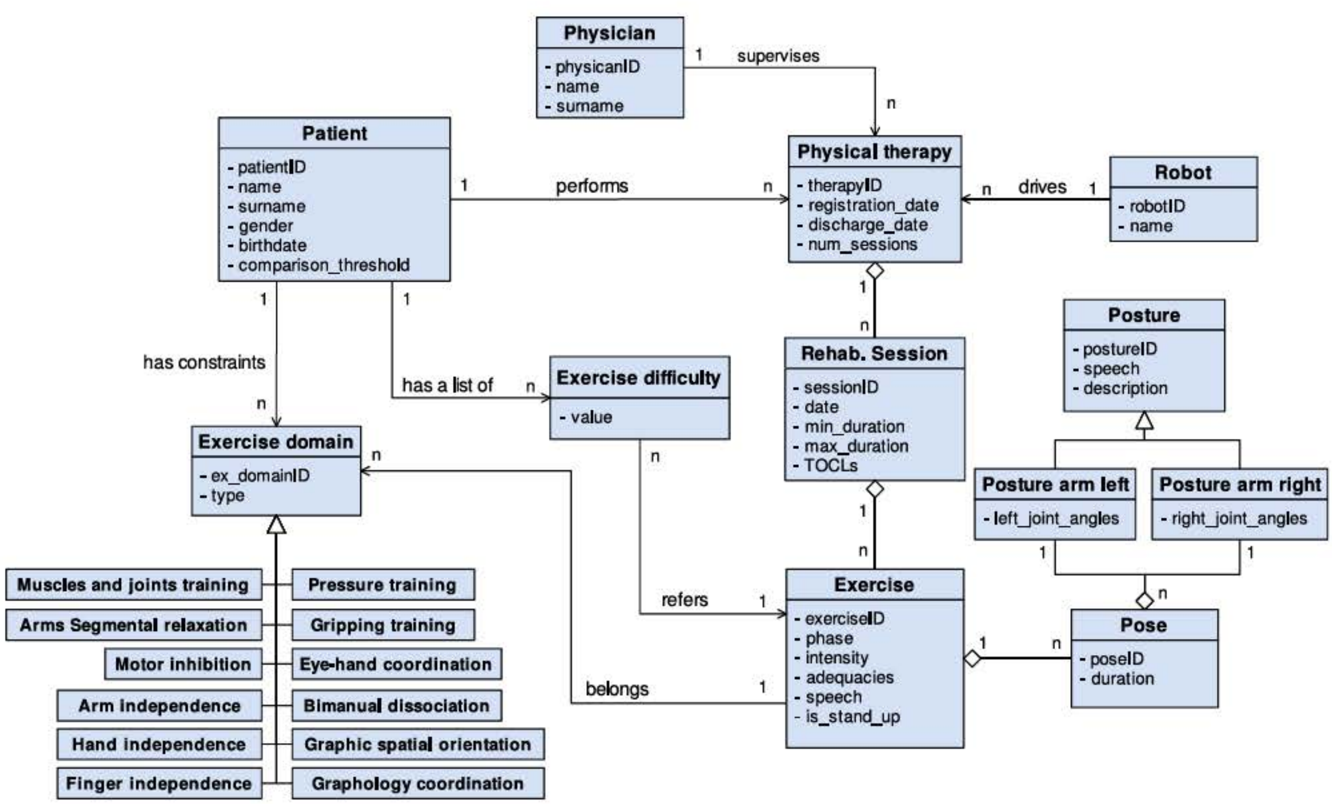

Fig. 4. NAOTherapist conceptual model.

of exercises can or cannot be trained by the patient and a value of difficulty for each exercise of the knowledge base. The attributes of a session are useful in determining whether the planned sessions meet the time constraints and the defined therapeutic objectives (TOCLs). It is crucial to have an enriched model of exercises that allows us to determine whether it contributes positively to the training of the patient or not. For this reason, attributes such as "adequacies" can be configured by the clinical experts. They represent a subjective numerical way of how well these exercises are appropriate to the therapeutic objectives of a session. The definition of the therapy and the information that is retrieved for this task is explained in detail in Section 4.

4. Therapy definition

Therapy Designer is a deliberative component based on Automated Planning that aims to generate therapy plans for patients with obstetric brachial plexus palsy and cerebral palsy. These high-level plans consist of a set of exercises which are then divided into a sequence of poses and subsequently executed by the robot. The system allows as many sessions as configured for the patient to be planned, therefore there may be extensive interactions among sessions due to the variability constraints. The high-level planning is also designed according to the clinical procedure of the HUVR and it is based on an internal guideline of this hospital for the rehabilitation of the targeted patients.

In order to configure the parameters of the therapies, all the information about the patient, sessions, exercises and poses is retrieved from the knowledge base associated to the conceptual model that is shown in Fig. 4 and explained in Section 3.2. The patient's constraints refer to those movements or exercises which may cause possible injuries or put the patient at risk. The capabilities are modeled as groups of exercises which can also be restricted to certain individuals. The difficulty and intensity of the conceptual model is a subjective numerical way of representing the exercise features based on the accumulated experience of therapists, that it is used to have a customized definition of the therapies. A rehabilitation session is organized as follows: the initial exercises are for warming up, the most intense ones are carried out in the middle of the session and the final phase is assigned to cooling down and relaxing exercises. Based on the results, the physician can update or refine new details of the therapy.

The therapeutic objectives are represented as cumulative levels which must be reached to achieve the planning goal. According to the clinical guidelines, the conceptual model considers five objectives to be trained: bimanual, fine unimanual, coarse unimanual, arm positioning and hand positioning activities. In the planning problem, these clinical objectives are modeled with five values which represent the training priorities that a patient has for each session. These objectives are called *Therapeutic Objectives Cumulative Levels* (TOCLs) and are established for each session, so they can be updated for future sessions in accordance with the progress of the patient. Achieving varied sessions is an important point to avoid disengagement and boredom while training. This feature is implicit in the model, so that there is a penalty for those exercises which have been previously included in other sessions.

The automatic therapy generation is correctly addressed in a hierarchical way due to the natural hierarchy of the problem. For this reason, an HTN approach is an appropriate technique to model the design of the therapies (Pulido et al., 2014). This proposal aims to provide a more easily extensible and configurable model in which expert knowledge can be included at any time. The methods and primitive actions of the hierarchical model are represented in Fig. 5, in which a therapy is a set of multiple sessions which in turn are broken down into three phases: warming up, training and cooling down. Each phase is completed with suitable exercises from the knowledge base according to its intensity and difficulty, which are expected to be distributed like a Gaussian-like function. The division between phases is given by axioms to represent the duration of each phase depending on the maximum and minimum time of the sessions. There are also axioms to decide the suitability of the exercises to each phase in order to decide whether they are candidates to be included or not. Fig. 6 shows the different numerical attributes that comprise the *e0* example exercise in HTN code. Two categories of attributes can be distinguished: (A) those which are related to the constraints of the problem and (B) those which refer to the TOCLs. Group A consists of the duration of exercises which is given in minutes, the intensity and difficulty established from 0 to 100 according to how tough the exercise is and the group of exercise referring to the associated trained capabilities. In the case of B attributes, they are the adequacy levels to the TOCLs, which are a representation of how well this exercise contributes to the therapeutic objectives. This contribution is defined as an integer from 0 to 3. The total contribution to the TOCLs in a session is calculated as the sum of all adequacy levels of the exercises included. Thus a valid therapy plan is one whose total contribution reaches the TOCLs established for the session. If there are no exercises available to be considered in knowledge base, the model allows a plan to be achieved in which it suggests creating or learning a new exercise whose attributes are planned according to the requirements of the session while ensuring the reachability of the TOCLs.

The planning algorithm follows the hierarchical decomposition while respecting the order relationships until reaching primitive actions. A therapy plan can comprise more than one session, so this is also considered in the hierarchical approach and represented in the model by a loop arrow (Fig. 5). Once the algorithm is in the process of completing a phase with exercises, the planner has to select those suitable according to the phase and variability constraints. However, this blind selection can be inefficient in more complex problems, in which TOCLs are tightly

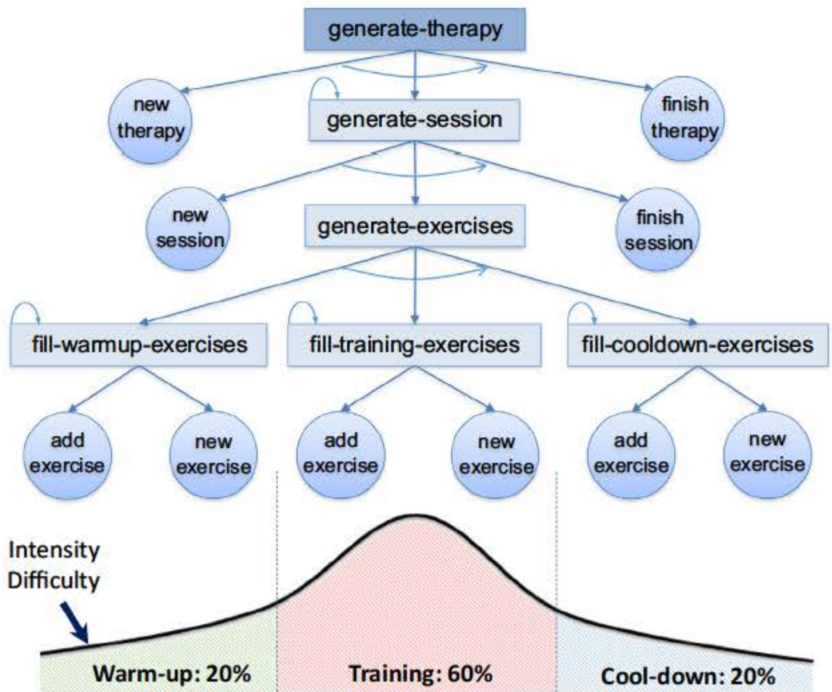

Fig. 5. The HTN model for therapy generation. Circles represent the primitive actions and rectangles refers to the methods of the model. The hierarchical decomposition is modeled with high to low arrows and the order relationships are represented with an horizontal curved arrow.

```
(exercise e0)
(e-duration e0 1.3)
(e-intensity e0 60)
(e-difficulty e0 70)
(e-group e0 g_train_muscles_joints)
(adqcy-bimanual e0 1)
(adqcy-fine-unimanual e0 0)
(adqcy-coarse-unimanual e0 1)
(adqcy-arm-positioning e0 2)
(adqcy-hand-positioning e0 0)
```

Fig. 6. Example of the HTN model of the exercise $e0$ retrieved from the knowledge base.

adjusted, since the total contribution of the exercises is not considered until reaching the total time of the session in the last phase. For this reason, a heuristic function (Eq. (1)) is proposed to drive the exercise selection process. This function returns a heuristic value that is calculated before every exercise inclusion. The first term of the summation evaluates the suitability of the exercises to the TOCLs, where d_i is the distance (minus operation) between the current cumulative level, assuming the exercise is included, to the established TOCLs for the planned session. The second part of the equation represents a penalty for the previously used exercises (ex_{times_used} is the number of times an exercise has been included in the set of $num_{sessions}$ sessions). So, the proposed function rewards those exercises whose contribution minimizes the distance to the frontier solution. This allows the selection of exercises to be driven to reduce the number of steps, instead of a blind selection which can cause many backtracking steps to find a valid plan.

$$ht_{ex} = \sum_{i=1}^{n_{objectives}} \left(\frac{1}{d_i^2 + 1} - \frac{ex_{times\_used}}{num_{sessions}} \right) \quad (1)$$

4.1. Evaluation of therapy designer

The automatic generation of therapies is addressed in a hierarchical way and belongs to the higher level of the architecture. In order to evaluate the performance of the HTN model, the JSHOP2 planner (Nau et al., 2003) was used for the experimentation, running in a PC with the following configuration: Intel Core i3, 3.30 GHz $\times$ 4, 8 GB of RAM. The first evaluation tries to demonstrate the performance of the therapy designer module in terms of planning time while increasing the complexity of the problems (Table 1). The second evaluation focuses on the therapeutic significance of the planned sessions. There are two experiments that validate, firstly the order of the exercises in the sessions under the clinical and variety criteria (Table 2), and secondly the obtained average distribution of the intensity and difficulty throughout the sessions (Fig. 7).

The planning process has three main goals: reaching the cumulative levels established by physicians (TOCLs), ensuring the variability of occurrences of exercises and

Table 1
Planning time in seconds, facing the blind selection against the proposed heuristic function for both (A) relaxed and (B) tightly adjusted experiments.

Selection	N. sessions					
	2	5	10	20	50	100
Exp. A						
Blind	1.50	1.74	2.70	4.74	13.21	30.75
Heuristic function	1.20	1.44	2.50	4.36	10.65	25.86
Exp. B						
Blind	1.02	8.66	>1800	>1800	>1800	>1800
Heuristic function	1.01	1.86	2.66	6.46	18.09	764.08

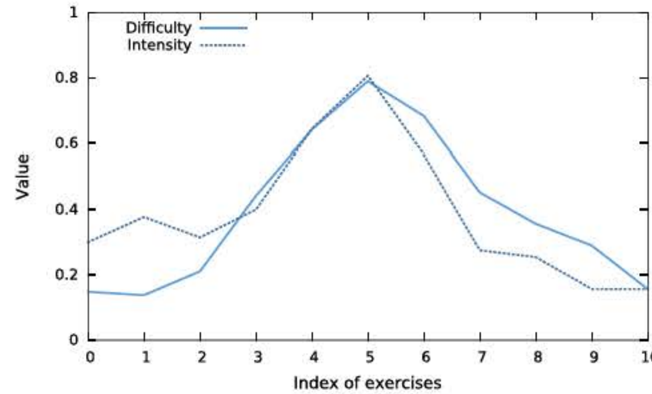

Fig. 7. Average values of intensity and difficulty with respect to the index of exercises of 30 generated sessions.

respecting the time limits of the session. The planning time is very dependent on the relationship between the TOCLs and time constraints. This means that problems with tightly-adjusted values require more time to find a suitable combination of exercises which achieves the TOCLs for the established session time. For this reason, two different configurations were evaluated to determine the performance of the heuristic function in contrast to the blind selection of exercises. It should be pointed out that blind policy is a circular selection by default in the planner. It is expected that the informed heuristic function reduces the number of inefficient bindings which may cause too many backtracks, affecting the performance.

Table 1 shows the results in seconds facing both selection policies while increasing the number of sessions to be planned. This was tested with 70 exercises in the knowledge base. Experiment *A* was carried out with a relaxed configuration of the problem. This means that TOCLs were low with respect to time constraints and exercises available. Although the time of the heuristic selection is low, the differences are not very significant. However, experiment *B* shows completely different results when the TOCLs are tightly adjusted. In this situation, the blind selection needs to try so many bindings to find a set of exercises that meets the established criteria. From the generation of 10 sessions, the time was more than 1800 s which is hardly acceptable when a quick response is expected.

Table 2 aims to show the distribution of 15 planned sessions with 70 exercises from the knowledge base. The session was configured with a duration that ranges from 25 to 30 min. The table has colored cells to represent the three phases that comprise a session (warm-up, training and cool-down). As can be seen, the penalty for repetition included in the heuristic function allows a variety of exercises between sessions and avoid cycles. The model also prevents the repetition of exercises in one session or in the same position as the last occurrence. In this case, since there are enough exercises, the model does not need to

Table 2
Distribution of the exercises of 15 executed sessions for the a patient with the same session objectives. The color of the cells represents the three phases of training: green for warm up, yellow for training and blue for cool down.

Sessions	Exercises																	
S1	e18	e19	e25	e26	e1	e11	e31	e29	e30	e17	e3	e10	e16	e4	e7	e9	e15	e6
S2	e12	e6	e22	e18	e0	e29	e30	e31	e8	e13	e14	e11	e17	e3	e9	e15	e27	
S3	e19	e22	e28	e25	e31	e2	e10	e16	e7	e8	e13	e14	e11	e9	e15	e12		
S4	e18	e19	e26	e27	e28	e1	e17	e20	e21	e24	e4	e3	e10	e16	e23	e22	e9	
S5	e6	e12	e18	e19	e25	e31	e5	e7	e8	e13	e14	e10	e3	e4	e9	e15	e22	
S6	e26	e27	e28	e25	e2	e0	e20	e31	e16	e3	e10	e4	e17	e11	e15	e6	e12	
S7	e18	e19	e26	e27	e28	e2	e11	e21	e20	e16	e4	e3	e10	e5	e22	e9	e15	e6
S8	e12	e6	e18	e19	e25	e31	e0	e8	e7	e14	e13	e16	e3	e10	e9	e15	e22	
S9	e26	e27	e28	e25	e1	e5	e21	e31	e29	e30	e16	e17	e7	e12	e6	e9		
S10	e18	e19	e26	e27	e28	e1	e17	e11	e8	e3	e10	e16	e13	e14	e7	e12	e15	e6
S11	e22	e18	e25	e26	e2	e11	e31	e29	e30	e20	e21	e24	e23	e19	e27	e28		
S12	e12	e6	e22	e18	e0	e29	e30	e31	e13	e8	e14	e17	e7	e9	e15	e25		
S13	e19	e22	e26	e27	e1	e17	e31	e20	e21	e5	e0	e4	e3	e10	e9	e15	e23	
S14	e28	e12	e6	e19	e20	e30	e11	e8	e14	e13	e7	e3	e16	e9	e15	e22		
S15	e18	e19	e25	e26	e5	e8	e29	e30	e31	e21	e20	e10	e4	e22	e9	e15		

Warm-up Training Cool-down

suggest new exercises, so that the planner is able to find a varied distribution of exercises that reaches the established therapeutic goals (TOCLs).

In order to evaluate the intensity and difficulty distribution, a problem with 72 exercises was solved with JSHOP2 with the following configuration: 30 sessions of 25 30 min each, 40% of the session time was divided evenly into warming up and cooling down and the remaining 60% was spent on the training phase. Those exercises whose intensity and difficulty are between 0 and 0.4 are considered by the model as warm-up or cool-down exercises, but when these values are greater than 0.4, they can be included in the training phase. The average intensity and difficulty value with respect to the index of exercises of the generated plans is shown in Fig. 7, where both distributions approximate a desired Gaussian-like function. With the aim of providing a customized shape of the function, the model considers that exercises can be more intense but less difficult in the warm-up phase and vice versa for the cool-down phase.

5. Session execution and monitoring

This section involves all three processes of the pipeline model of cognition. It describes the use case of NAOTherapist and then explains the reasoned deliberation of medium-level actions according to the perceived state of the world. Five components of the architecture (Fig. 3) are involved in this task: Decision Support, Executive, Vision, Kinect Sensor and Robot.

5.1. Use case and session requirements

For the definition of the rehabilitation sessions, the execution flow of Fig. 8 has been designed according to the use case of the project and hospital requirements. The first and last rows of boxes refer to the welcome and parting interactive stage. The middle rows represent the training stage in which the robot and patient perform the exercises together.

In dynamic environments like these, unexpected situations will occur and a reasoned answer is necessary. For instance, if the patient is distracted or decides to leave the training area, instead of continuing with the training, the system may claim the attention of the patient to recover their focus. If the robot is overheating or its battery is low, the system could detect this situation and execute the appropriate actions to cool the motors down or ask for its battery to be recharged. In this way, the robotic platform is able to behave autonomously. Furthermore, ensuring that the patient performs the exercises correctly is an essential requirement of this work. At the time of training, the system checks whether the angles of the joints of the patient correspond with the poses performed by the robot. If the poses differ, the robot warns the user and shows them how to correct it. It is also very important to send motivational speech to the patient and congratulate him when he is doing it correctly.

The use case[3] starts when the patient enters the experimental room and finds the robot placed in the demonstration area. Then, the system tracks the patient and starts capturing his body characteristics. The patient is one or two meters away from the robot in the training area. The robot greets him and the training begins after introducing the first exercise. The exercises are made up of a sequence of poses. Depending on the exercise, the patient must maintain each pose for a certain amount of time. The robot is in charge of driving the training process giving instructions and feedback on what to do at each time. Each pose of the patient is verified with respect to that shown by the robot. If both poses differ, the system executes a correction mechanism. Patients have two attempts performing a pose correctly: after the first failed attempt, the robot shows the incorrect arm or arms and tells the patient that the pose must be corrected. In the second correction, the robot imitates the detected posture of the patient and shows him how to move the arms to achieve the correct pose. This is called "mirrored correction". These mechanisms provide helpful feedback to users and help them to get closer to the correct pose. If the patient fails after these two tries, the pose is skipped. The system runs the rest of poses that comprises the exercise sequentially until it finishes. A break is programed between exercises to have a rest. In these pauses, the robot may show animations, choreography or tell stories to energize each break. Once all of the exercises are completed, the training is finished. The robot closes the session with a cheerful farewell, inviting him to play with him again the next day.

The autonomy of the robot is an important feature of the usefulness of the system for the human therapist, but it is not intended to serve as a way of replacing him in the rehabilitation sessions. NAOTherapist has been designed as a support tool for the therapist.

The architecture fulfills all requirements of the aforementioned use case, but the use of Automated Planning also makes it easier to change the domain to achieve different therapeutic goals or even different contexts away from the medical model. We tested this flexibility by using the same architecture with an adapted Simon game with poses instead of colors (Turp, Pulido, González, & Fernández, 2015), in which the robot performs several poses in a row and the user has to memorize and perform them to advance to longer rounds.

5.2. Session planning and monitoring

As already mentioned in previous sections, each session comprises a sequence of previously planned exercises. However, the output of the high-level planner indicates only the order of the exercises that must be performed. To control the session, NAOTherapist uses a lower level mechanism of planning (medium level in this case).

[3] Video of the use case: https://youtu.be/75xb39Q8QEg.

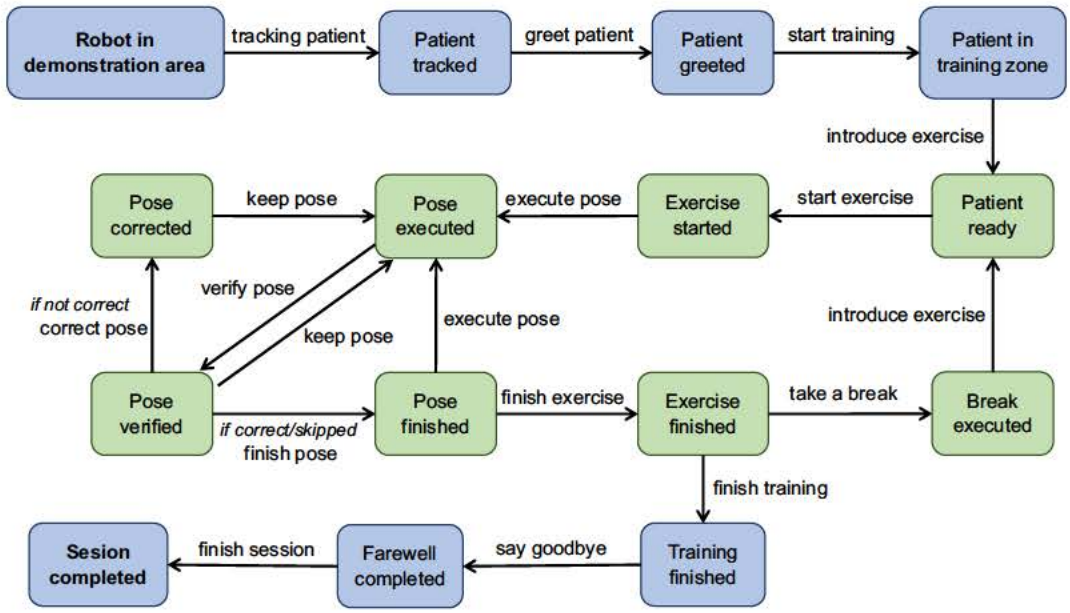

Fig. 8. Execution flow of NAOTherapist use case.

Medium-level planning includes the PELEA planning and replanning sub-architecture (Alcázar et al., 2010) integrated into the Decision Support component. It must work in real time because its response has to be fast enough to avoid delays in the execution of the therapy, allowing a fluent interaction. This includes all communications between the different components of the architecture to avoid bottlenecks. It combines three elements: monitoring, decision support and execution. This scheme matches the general pipeline of cognition of the cognitive architectures: perception, cognition and action.

5.2.1. Perception

Fig. 9 shows the different elements used to control the execution of the session. The hardware in charge of the perception of the state of the world is the 3D Kinect sensor and the robot for propioception data. The memory of our system is inside the Executive component, which gathers all information perceived by these elements to reproduce the actual state of the world. This state is made up of symbolic information in PDDL format, specifically predicates and functions. The information is divided into two types, depending on its origin: whether it comes from the environment (exogenous predicates) or not. The information from the environment is perceived by sensors and can change unexpectedly, while the other has control data that will always have the expected values.

The information perceived by the Kinect is analyzed by the Vision component. One of its functions is to perform the pose comparison to calculate how correct the pose that the patient is doing is. The method calculates differences between the expected pose and the pose of the patient and it is sent to Execution. This component uses a dynamic threshold adapted to each patient to determine whether a posture is valid or not. Vision also implements a situation awareness system to take into account situations that must be detected in order to maintain a coherent interaction with the patient. For instance, the patient leaves the training area, sits down or stops doing the exercises. The robot must be able to perceive certain parameters about himself like the battery level, joint temperature or whether it is standing or sitting. This information is sent to the Executive component to update the corresponding exogenous predicates.

5.2.2. Cognition

The decision making process about what the next action to be performed by the system is, is made using Automated Planning with the Decision Support component. These medium-level actions are similar to those shown in the use case (Fig. 8). When Executive needs a new one, it sends the actual state of the world to this component and Decision Support returns it the next action. Decision Support integrates a version of the PELEA sub-architecture for this purpose, which in NAOTherapist consists of three modules: Executive, Monitoring and the Planner. The Executive module of PELEA communicates with the Execution component of NAOTherapist. When it receives a new actual state of the world, it is sent to Monitoring to be compared with an expected state of the world, generated internally by this module. This is necessary because the exogenous predicates can vary unexpectedly, invalidating the previously generated plan. If the actual and the expected state of the world are compatible (whether the effects of the last action and the preconditions of the next one are in the actual state), then the next previously planned action is returned. If their differences invalidate the previous plan, the Planner module (MetricFF in this case (Hoffmann, 2003)) is executed again to obtain a new plan, taking the last actual state of the world as the initial state.

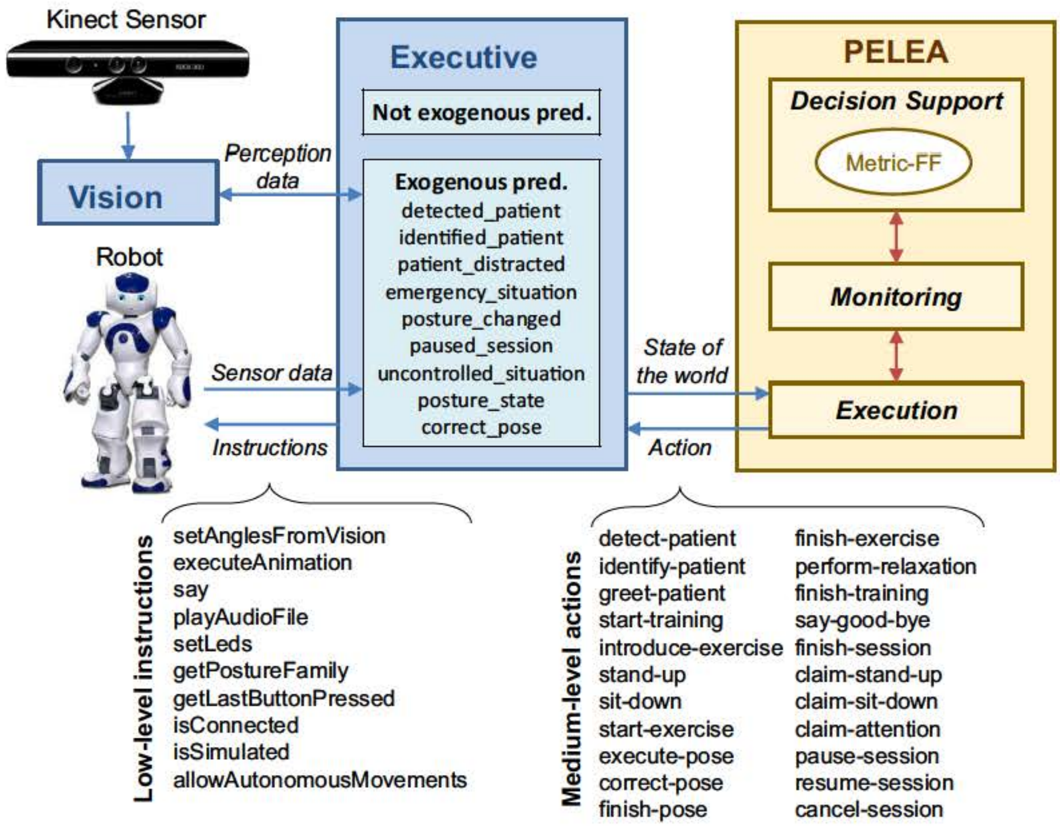

Fig. 9. Planning medium level actions with PELEA sub architecture and translation of these actions into low level instructions to the robot while preserving the state of the world.

Automated Planning is much more versatile than a state machine, but it is not as fast because it requires a search process. However, the obtained planning times are small enough to ensure a fluent interaction. Fig. 10 shows the average response time of the medium-level planning when returning a replanned action (+) and a previously planned action (×). Time measurement starts when the Executive component asks for an action and finishes when it receives the next action from Decision Support. This measure includes all communication delays. For this plot, 62 short sessions of 23 poses performed with healthy children were considered. All of them were executed in a laptop with an Intel Core i7 2.20 GHz × 8 and 8 GB of RAM.

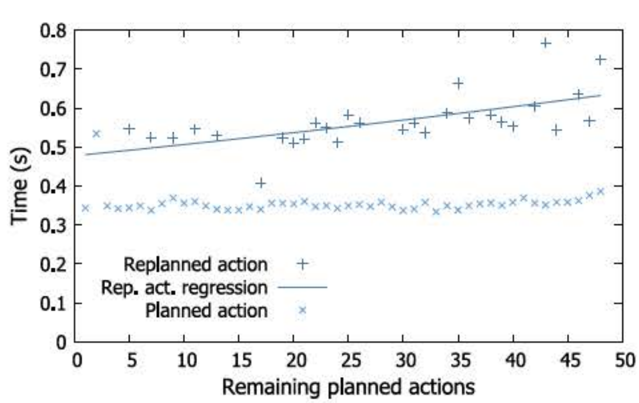

Fig. 10. Average time to receive the next action from the Decision Support with and without replanning.

While the plan continues to be valid and the Decision Support component returns a previously planned action, the response time is stable at 0.35 s. The processing of the action by the Executive component and the start of the action by the robot is almost immediate. Each action takes a certain time to be executed, depending on the corresponding animation. When the previous plan is no longer valid and the next action has to be replanned, response times continue to be very low, at about 0.55 s. When there are more actions to be executed, replanning will take more time following an exponential regression curve. However, in our system, planning is fast enough to hardly detect this increase, as can be seen in the flatness of the regression curve.

In all cases, the generated plans are correct and follow the lines previously established by the high-level planning. These results show that actions take hardly any time to be decided, so this allows a fluent rehabilitation session to be carried out.

5.2.3. *Action*

Once Decision Support has returned the next medium-level action to execute (by replanning or not), the Execution component subdivides it into several sequential or parallel low-level instructions that are sent to the robot, as shown in Fig. 9. The Robot component interprets these

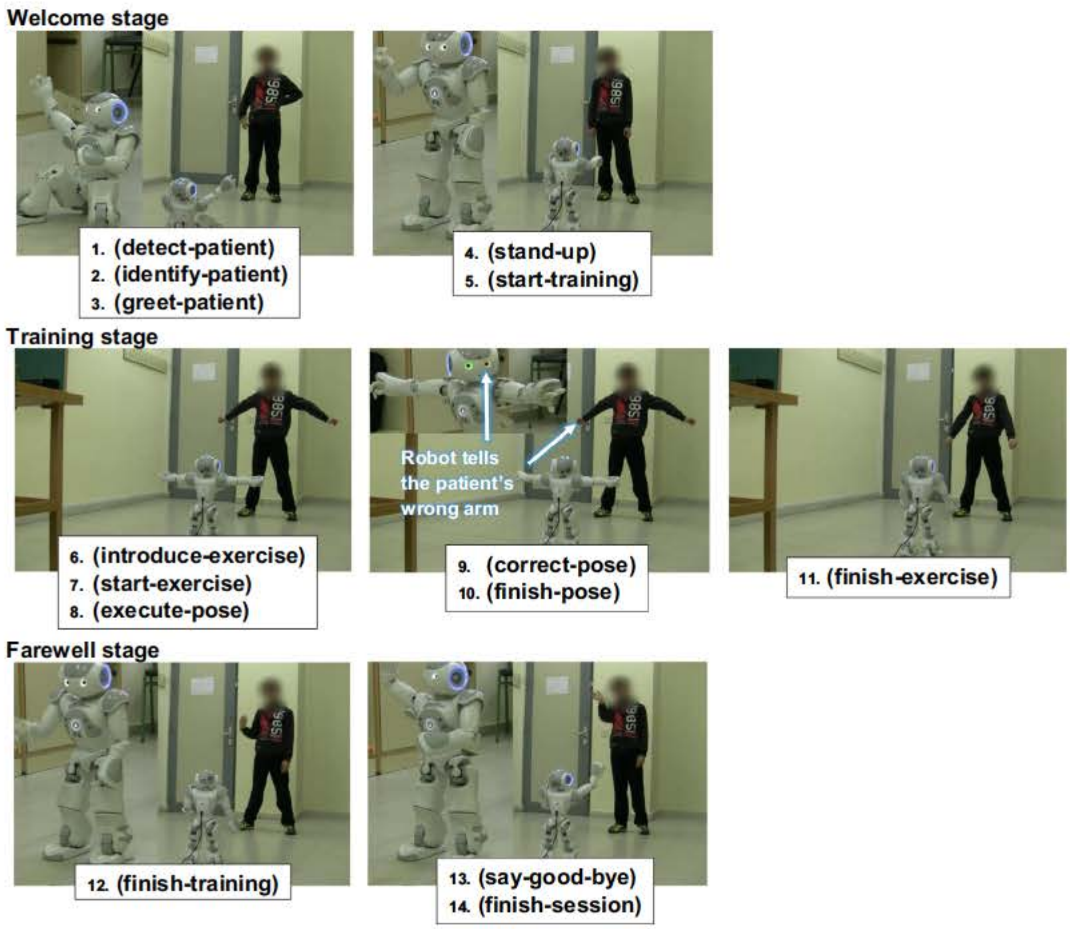

Fig. 11. Highlights of a video sequence with the NAO robot, controlled by the NAOTherapist architecture, and a child performing the basic use case. The medium level PDDL actions are depicted below each frame.

low-level instructions for the robotic platform that is being used and consequently acts.

Fig. 11 shows highlights of a session video in which a NAO robot is performing the use case described in Section 5.1 with a child. The required and ordered PDDL actions are detailed for each frame and for each interactive stage (welcome, training, farewell). A brief explanation of each medium-level action is provided in Appendix A.

5.3. Robot generic interface

Although NAOTherapist is designed to exploit the capabilities of the NAO robot in mind, one major design criterion is also to maintain independence from the robotic platform. To achieve this, the robot is controlled using a generic RoboComp interface (the Robot component) shared with the Executive component. We tested the architecture on three different kinds of humanoid robots with different capabilities[4]: a NAO robot, the Ursus robot and a REEM/REEM-C robot, as is shown in Fig. 12. For the first robot, the evaluation is made on the physical robot, while for the others a simulator is used (RCIS for Ursus and Gazebo for REEM).

The movement interpolation between robot poses is carried out in a low-level planner behind this generic interface. The task of low-level path planning is managed by each robot, so the evaluation of response time is out of the scope of this paper.

Robot receives each low-level instruction from Execution and orders the robot to perform it. To perform the poses, the arm positions are stored in the knowledge base of the system as Kinect skeleton models to make them independent of the robots. The Robot component receives these skeleton models and must perform a re-targeting process or adaptation from this skeleton to the configuration of the joints of its particular robot.

In essence, the NAO robot can be substituted by any other just by making another generic interface adapted to it. The new robot should have the same capabilities required for NAOTherapist (arms, speakers, lights, feedback, etc.), but if it lacks some of them, its interface must at least return a coherent response to every low-level instruction of Executive to continue with the rehabilitation

[4] Playlist with videos of the tested robotic platforms: https://goo.gl/FKVsqm.

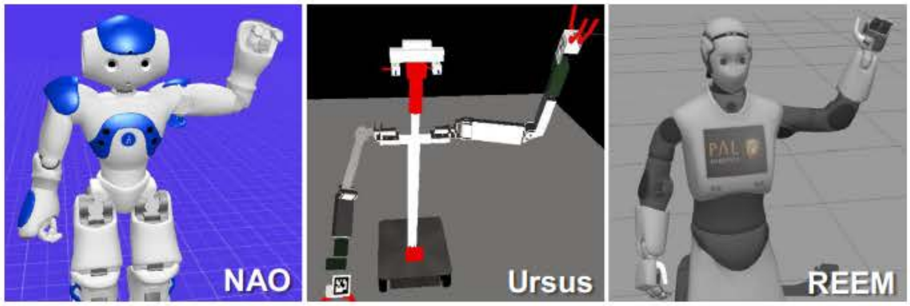

Fig. 12. Demonstration of the retargeting of a posture (stored as a Kinect skeleton model) with three different robots.

session. Moreover, features of new robotic platforms can be customized using the generic interface of the Robot component without affecting the main flow of actions of the architecture.

There are several generic low-level instructions to control the robot and get information from it (Fig. 9). All instructions can be blocking (freezing the execution until the instruction finishes completely) or non-blocking, allowing the execution of several in parallel. They are summarized in Appendix B.

This allows a lot of flexibility in some unexpected aspects. For example, the eye-colored feedback of the NAO robot is useful for showing patients how well they are performing the poses. The eyes of NAO become a more intense green when the pose is completely correct. This mechanism cannot be reproduced by the REEM robot because it does not have LEDs in the eyes. To solve this, a graphical interface is displayed on the touchscreen of its chest with equivalent feedback (bars filling when the pose is more correct, circles fitting a mark or faces smiling more or less).

This independence from the architecture also has advantages beyond compatibility. Simple low-level instructions like "say" can be reprogrammed to add more functionality very easily. For instance, speech can be played by different text-to-speech engines or using pre-recorded voices. When the low-level instruction requires something to be said with a pre-recorded voice, the Robot component tries to find the sound file with the name that matches the text to be said. If the sound file is not located, then the sentence can be synthesized.

6. Conclusion

Within a general framework of hands-off robotics rehabilitation, this paper describes the design, deployment and initial evaluation of a novel software architecture. Following the general pipeline cognition model, the NAOTherapist architecture allows a humanoid robot to drive therapeutic sessions previously planned by the system autonomously. The control is addressed at three different abstraction levels, from the higher one, in which the whole therapy is planned, to the lower one, in which the robot moves.

The whole architecture has been extensively evaluated with a large group of schoolchildren[5] and it is currently working with pediatric patients in a real-case scenario under the supervision of the clinical experts of the HUVR.[6] The results of these evaluations are an ongoing work and they are out of the scope of this paper. We could advance, according to these preliminary results, that the robotic system was well received by schoolchildren, patients and professionals alike. Patients liked to follow the exercises proposed by the robot and they were engaged with the therapy making an effort to perform the poses. The robot was also able to carry out the sessions autonomously without the need for human intervention. The correct integration of all components has been successfully tested, so the NAOTherapist project has a very polished and reliable prototype.

The application of this prototype in actual therapies is possible. Physicians found that the features provided by NAOTherapist could be very useful to ease their workload. The NAO robot could be affordable for some hospitals or other rehabilitation institutions and it is resistant enough to withstand repeated falls. The independence achieved from the robot and the application domain could allow the use of other low-cost robots to make NAOTherapist affordable for families at home too, as well as bigger robots for adult patients.

Before we can perform large-scale experiments it is important to design and endow the robot with more complex interactive channels so that children can become more engaged. This can be done by using additional resources (e.g. conversational abilities or a larger collection of games and activities).

In a near future, we are planning to merge NAOTherapist within the RoboCog architecture of the Therapist project (Calderita et al., 2014) to adapt some of its components which could extend the perception capabilities: the speech, activity, emotion and face recognition components, for instance. The inclusion of these new elements enriches the state of the world, allowing their use in the planning phases to improve the behaviors and the autonomy of the robot.

[5] Evaluation of the interaction with schoolchildren: https://youtu.be/Dl4MwWg7PD4.

[6] Initial evaluation with pediatric patients in HUVR: https://youtu.be/acopdNtddlM.

Appendix A. Medium-level action descriptions

Each medium-level action is planned by the Decision Support component which is translated into generic instructions by the Executive component to be executed by the robot:

- detect-patient: The system tracks the patient and starts serving his anthropometric data.
- identify-patient: The system loads the profile of the patient.
- greet-patient: The robot executes a greeting animation and plays a welcome speech.
- start-training: The robot introduces the ongoing activity to the patient.
- introduce-exercise: The robot gives a short explanation of the next exercise.
- stand-up: The robot stands up.
- sit-down: The robot sits down.
- start-exercise: is a control action that prepares internally the system for the upcoming exercise.
- execute-pose: The Executive component sends the pose to be imitated with both arms to the robot. The path planning of the robot is in charge of planning the movement interpolation at a low level. The system starts comparing the pose. Each pose must be correctly maintained as long as that is indicated in the exercise.
- correct-pose: It is executed if the last pose has not been performed correctly or it has not been maintained for the required amount of time. In the first correction, the robot indicates the wrong arm to correct. In the second correction, the robot imitates the detected posture of the patient and shows him how to move the arms to achieve the correct pose. If the patient fails in these two corrections, the pose is skipped.
- finish-pose: It terminates the pose and prepares the system for the upcoming pose.
- finish-exercise: The robot notifies the patient that the current exercise is complete and encourages him to continue with the training.
- finish-training: The robot executes animations and speech to notify the patient that the training is finished for today.
- perform-relaxation: This action is used to take a break between exercises by encouraging the patient to breathe deeply for recovery or showing animations, choreography, etc.
- say-good-bye: The robot says good-bye.
- finish-session: This action represents the end of the session, where the robot takes the initial rest posture to wait for the next user.
- claim-stand-up: If the patient is seated and the exercise requires him to be standing, the robot asks the patient to stand up.
- claim-sit-down: If the patient is standing and the exercise requires him to be seated, the robot asks the patient to sit down.
- claim-attention: If the Vision component detects that the patient is distracted, the robot attracts his attention.
- pause-session: The session is paused and the system allows the therapist to see why.
- resume-session: This is triggered by the therapist and allows the rehabilitation to continue.
- cancel-session: This is triggered by the therapist cancelling the session.

Appendix B. Low-level instruction descriptions

A set of low-level instructions are executed by the Robot component for each medium-level action:

- setAnglesFromVision: It receives a Kinect skeleton model of the upper body and sets the angles of the arms of the robot accordingly through a retargeting process.
- executeAnimation: It executes a predefined animation recorded for each particular robot or transformed from other robot. It can involve several aspects such as LEDs, speech and movement together. The instruction receives the name of the animation and each robot is completely free to interpret it in its own way.
- say: It receives text to be said by the robot. A parameter controls whether the text is synthesized or stored in a pre-recorded file. If the file is not available for this particular speech, then the text is synthesized.
- playAudioFile: Plays an audio file like music or effects, not necessarily related to speech.
- setLeds: This instruction receives a string with generic led groups to modify their intensity to give visual feedback to the user. Depending on each robot, these groups can exist or not, so each one can interpret this instruction in its own way, using an internal screen for example to simulate them.
- getPostureFamily: Checks whether the robot is standing, sitting, etc. If the robot does not have legs, the posture will be always standing.
- getLastButtonPressed: Stores the last button pressed on the robot. The architecture manages some generic buttons to perform basic external control for the therapist, like interrupting the session. These buttons could be in the robot, on a touchscreen, in an external user interface controlled in a computer, etc.
- isConnected: Checks whether the robot is fully started.
- isSimulated: Checks whether the robot is simulated or not.
- allowAutonomousMovements: Some robots have an internal ability to carry out small movements by themselves to appear more “organic” to the users. This function disables these movements when performing a pose and re-enables them when performing social interaction.

References

Alcázar, V., Guzmán, C., Prior, D., Borrajo, D., Castillo, L., & Onaindia, E. (2010). PELEA: Planning, learning and execution architecture. In *Proceedings of the 28th workshop of the UK Planning and Scheduling Special Interest Group (PlanSIG). Brescia, Italy*.

Avery, E., Kelley, T., & Davani, D. (2006). Using cognitive architectures to improve robot control: Integrating production systems, semantic networks, and sub symbolic processing. In *Proceedings of 15th annual conference on Behavioral Representation in Modeling and Simulation (BRIMS)*.

Baxter, P. E., de Greeff, J., & Belpaeme, T. (2013). Cognitive architecture for humanrobot interaction: Towards behavioural alignment. *Biologically Inspired Cognitive Architectures, (6)*, 30 39 (pp. 30 39).

Benjamin, P., Lyons, D., & Lonsdale, D. (2004). ADAPT: A cognitive architecture for robotics. In *Proceedings of the 6th International Conference of Cognitive Modeling (ICCM)*.

Boccanfuso, L., & O'Kane, J. M. (2011). Charlie: An adaptive robot design with hand and face tracking for use in autism therapy. *International Journal of Social Robotics, (3)*, 337 347 (pp. 337 347). http://dx.doi.org/10.1007/s12369 011 0110 2.

Borggraefe, I., Kiwull, L., Schaefer, J. S., Koerte, I., Blaschek, a., Meyer Heim, a., & Heinen, F. (2010). Sustainability of motor performance after robotic assisted treadmill therapy in children: An open, non randomized baseline treatment study. *European Journal of Physical and Rehabilitation Medicine, 46*, 125 131.

Brisben, A. J., Safos, C. S., Lockerd, A. D., Vice, J. M., & Lathan, C. E. (2005). The CosmoBot™ system: Evaluating its usability in therapy sessions with children diagnosed with cerebral palsy. Technical Report AnthroTronix, Inc. URL <http://web.mit.edu/zoz/Public/AnthroTronix ROMAN2005.pdf> visited on 2016 07 25.

Cabibihan, J. J., Javed, H., Ang, M., & Aljunied, S. M. (2013). Why robots? A survey on the roles and benefits of social robots in the therapy of children with autism. *International Journal of Social Robotics, 5*, 593 618. http://dx.doi.org/10.1007/s12369 013 0202 2.

Calderita, V. L., Manso, J. L., Bustos, P., Suárez Mejías, C., Fernández, F., & Bandera, A. (2014). THERAPIST: Towards an autonomous socially interactive robot for motor and neurorehabilitation therapies for children. *JMIR Rehabilitation and Assistive Technologies (JRAT), 1*, e1. http://dx.doi.org/10.2196/rehab.3151.

Dehkordi, P. S., Moradi, H., Mahmoudi, M., & Pouretemad, H. R. (2015). The design, development, and deployment of roboparrot for screening autistic children. *International Journal of Social Robotics, 7*, 513 522. http://dx.doi.org/10.1007/s12369 015 0309 8.

Dematteo, C., & Pollock, N. (1992). Quality of upper extremity skills test. *Physical Occupational Therapy in Pediatrics, 13*, 1 18.

Dickinson, H. O., Parkinson, K. N., Ravens Sieberer, U., Schirripa, G., Thyen, U., Arnaud, C., ... Michelsen, S. I. (2007). Self reported quality of life of 8 12 year old children with cerebral palsy: A cross sectional european study. *The Lancet, 369*, 2171 2178.

Drubicki, M., Rusek, W., Snela, S., Dudek, J., Szczepanik, M., Zak, E., ... Sobota, G. (2013). Functional effects of robotic assisted locomotor treadmill thearapy in children with cerebral palsy. *Journal of Rehabilitation Medicine: Official Journal of the UEMS European Board of Physical and Rehabilitation Medicine, 45*, 358 363. http://dx.doi.org/10.2340/16501977 1114.

Fasola, J., & Mataric, M. (2010). Robot exercise instructor: A socially assistive robot system to monitor and encourage physical exercise for the elderly. In *RO MAN, 2010 IEEE* (pp. 416 421). http://dx.doi.org/10.1109/ROMAN.2010.559865.

Feil Seifer, D., & Mataric, M. J. (2005). Defining socially assistive robotics. In *Proceedings of the 9th International Conference on Rehabilitation Robotics (ICORR)* (pp. 465 468). IEEE.

Fox, M., & Long, D. (2003). PDDL2.1: An extension to PDDL for expressing temporal planning domains. *Journal of Artificial Intelligence Research (JAIR), 20*, 61 124.

Galindo, C., Gonzalez, J., & Fernandez Madrigal, J. A. (2005). An architecture for cognitive human robot integration. Application to rehabilitation robotics. *2005 IEEE International Conference Mechatronics and Automation* (Vol. 1, pp. 329 334). . http://dx.doi.org/10.1109/ICMA.2005.162656.

Ghallab, M., Nau, D., & Traverso, P. (2004). *Automated Planning: Theory & Practice*. Elsevier.

González, J. C., Pulido, J. C., Fernández, F., & Suárez Mejías, C. (2015). Planning, execution and monitoring of physical rehabilitation therapies with a robotic architecture. In *Proceedings of the 26th Medical Informatics Europe conference (MIE). Studies in Health Technology and Informatics* (Vol. 210, pp. 339 343). European Federation for Medical Informatics (EFMI) Madrid (Spain): IOS Press. http://dx.doi.org/10.3233/978 1 61499 512 8 339.

Gross, H. M., Debes, K., Einhorn, E., Mueller, S., Scheidig, A., Weinrich, C., ... Martin, C. (2014). Mobile robotic rehabilitation assistant for walking and orientation training of stroke patients: A report on work in progress. In *2014 IEEE International Conference on Systems, Man and Cybernetics (SMC)* (pp. 1880 1887). http://dx.doi.org/10.1109/SMC.2014.697419.

Hoffmann, J. (2003). The metric FF planning system: Translating "Ignoring Delete Lists" to numeric state variables. *Journal of Artificial Intelligence Research (JAIR), 20*, 291 341.

Krigger, K. W. (2006). Cerebral palsy: An overview. *Am Fam Physician, 73*, 91 100.

Malik, N. A., Hanapiah, F. A., Rahman, R. A. A., & Yussof, H. (2016). Emergence of socially assistive robotics in rehabilitation for children with cerebral palsy: A review. *International Journal of Advanced Robotic Systems, 13*.

Manso, L., Bachiller, P., Bustos, P., Nez, P., Cintas, R., & Calderita, L. (2010). RoboComp: A tool based robotics framework. In N. Ando, S. Balakirsky, T. Hemker, M. Reggiani, & O. von Stryk (Eds.), *Simulation, modeling, and programming for autonomous robots. Lecture notes in computer science* (Vol. 6472, pp. 251 262). Berlin, Heidelberg: Springer. http://dx.doi.org/10.1007/978 3 642 17319 6 25.

Martín, A., González, J. C., Pulido, J. C., García Olaya, A., Fernández, F., & Suárez, C. (2015). Therapy monitoring and patient evaluation with social robots. In *Proceedings of the 3rd workshop on ICTs for improving Patients Rehabilitation Research Techniques (REHAB)* (pp. 152 155). Lisbon (Portugal): ACM. http://dx.doi.org/10.1145/2838944.283898.

Mataric, M., Eriksson, J., Feil Seifer, D., & Winstein, C. (2007). Socially assistive robotics for post stroke rehabilitation. *Journal of NeuroEngineering and Rehabilitation, 4*. http://dx.doi.org/10.1186/1743 0003 4 5.

Mead, R., Wade, E., Johnson, P., Clair, A. S., Chen, S., & Matarić, M. J. (2010). An architecture for rehabilitation task practice in socially assistive human robot interaction. In *Proceedings of the 19th international symposium on robot and human interactive communication*.

Nalin, M., Baroni, I., & Sanna, A. (2012). A motivational robot companion for children in therapeutic setting. In *IROS 2012*.

Nau, D., Au, T. C., Ilghami, O., Kuter, U., Murdock, J. W., Wu, D., & Yaman, F. (2003). SHOP2: An HTN planning system. *Journal of Artificial Intelligence Research (JAIR), (20)*, 379 404 (pp. 379 404).

Ng Thow Hing, V., Thorisson, K. R., Sarvadevabhatla, R. K., Wormer, J. A., & List, T. (2009). Cognitive map architecture. *IEEE Robotics Automation Magazine, 16*, 55 66. http://dx.doi.org/10.1109/MRA.2008.931634.

Prenzel, O., Feuser, J., & Graser, A. (2005). Rehabilitation robot in intelligent home environment software architecture and implementation of a distributed system. In *Proceedings of the 9th International Conference on Rehabilitation Robotics (ICORR)*.

Pulido, J. C., González, J. C., González Ferrer, A., García, J., Fernández, F., Bandera, A., ... Suárez, C. (2014). Goal directed generation of exercise sets for upper limb rehabilitation. In *Proceedings of Knowledge Engineering for Planning and Scheduling workshop (KEPS), ICAPS* (pp. 38 45).

Ramos, L. E., & Zell, J. P. (2000). Rehabilitation program for children with brachial plexus and peripheral nerve injury. *Seminars in pediatric neurology* (Vol. 7, pp. 52 57). Elsevier.

Šabanović, S., Bennett, C. C., Chang, W. L., & Huber, L. (2013). Paro robot affects diverse interaction modalities in group sensory therapy for older adults with dementia. In *2013 IEEE International Conference on Rehabilitation Robotics (ICORR)* (pp. 1 6). IEEE.

Shamsoddini, A., Amirsalari, S., Hollisaz, M. T., Rahimniya, A., & Khatibi Aghda, A. (2014). Management of spasticity in children with cerebral palsy. *Iranian Journal of Pediatrics, 24*.

Suárez Mejías, C., Echevarría, C., Nuez, P., Manso, L., Bustos, P., Leal, S., & Parra, C. (2013). Ursus: A robotic assistant for training of children with motor impairments. In *Converging clinical and engineer ing research on neurorehabilitation*. *Biosystems & biorobotics* (Vol. 1, pp. 249 253). Berlin, Heidelberg: Springer. http://dx.doi.org/10.1007/978 3 642 34546 3 3.

Sun, R. (2001). *Duality of the mind A bottom up approach toward cognition*. Lawrence Erlbaum.

Tapus, A., Mataric, M., & Scasselati, B. (2007). Socially assistive robotics [Grand Challenges of Robotics]. *IEEE Robotics Automation Magazine, 14*, 35 42. http://dx.doi.org/10.1109/MRA.2007.339605.

Tapus, A., Tapus, C., & Mataric, M. J. (2009). The use of socially assistive robots in the design of intelligent cognitive therapies for people with dementia. In *2009 IEEE international conference on rehabilitation robotics* (pp. 924 929). http://dx.doi.org/10.1109/ICORR.2009.520950.

Trafton, J. G., Harrison, A. M., Fransen, B., & Bugajska, M. (2009). An embodied model of infant gaze following. In *Proceedings of the 9th International Conference of Cognitive Modeling (ICCM)*.

Turner Stokes, L. (2009). Goal attainment scaling (GAS) in rehabilitation: A practical guide. *Clinical Rehabilitation, (23)*, 362 370 (pp. 362 370). http://dx.doi.org/10.1177/0269215508101742.

Turp, M., Pulido, J. C., González, J. C., & Fernández, F. (2015). Playing with Robots: An Interactive Simon Game. In *Proceedings of the 16th Conference of the Spanish Association for Artificial Intelligence (CAEPIA), RSIM workshop, Albacete (Spain)* (pp. 1085 1095).