# Goal-directed Generation of Exercise Sets for Upper-Limb Rehabilitation Assisted by Humanoid Robots

**José Carlos Pulido, José Carlos González, Arturo González-Ferrer, Javier García, Fernando Fernández**
Planning and Learning Group, Departamento de Informática
Universidad Carlos III de Madrid

**Antonio Bandera**
Departamento de Tecnología Electrónica
Universidad de Málaga

**Pablo Bustos**
Robotics and Artificial Vision Laboratory
Universidad de Extremadura

### Abstract

A rehabilitation therapy usually derives from general goals set by the medical expert, who requests the patient to attend sessions during a certain time period in order to help him regaining mobility, strength and/or flexibility. The therapist must transform these general goals manually into a set of exercises distributed over different rehabilitation sessions that compose the complete therapy plan, taking into account the patient clinical conditions and a predetermined session and therapy time. This becomes a hard task and might lead to rigid schedules which not always accomplish the desired achievement level of therapeutic objectives established by the physician and could have a negative impact on the patients' engagement in the therapy. In this paper we present a method based on Automated Planning for the automatic generation of therapy plans for patients suffering obstetric braquial plexus palsy, in response to a given set of therapy goals. The ultimate purpose is to project the therapy plans to robots that can help patients achieving a better performance, showing them how to do exercises properly.

## Introduction

Clinical Decision Support Systems (CDSS) have been developed in the last decades to facilitate many tasks of physicians, like helping them in implementing Clinical Practice Guidelines (CPGs) through ad-hoc computer-interpretable models (Peleg 2013). In some cases, it might happen that the protocol to treat a patient condition is not so clear, and that the procedure to design the treatment pathway depends directly of a set of expected therapeutic objectives that the patient should achieve. In this case, guidelines can only give high-level recommendations on what combination of therapies to establish for a patient condition, but it still may require high complexity for the physician to deal with the configuration of the most appropriate combination of steps to maximize the expected outcome, for example according to a standard scale. This is the case of rehabilitation therapies for obstetric braquial plexus palsy (OBPP), the condition where this paper is scoped. OBPP is a serious injury that causes a loss of movement or weakness of the affected upper-limb. It is produced when the collection of nerves around the shoulder are damaged during the birth. This kind of complication has been reduced due to the improvements of the birth process but still about 1,5 of every 1.000 live births present this injury and require physical therapy. In order to design the OBPP rehabilitation stage in the Virgen del Rocío University Hospital (Seville, Spain)[1], a set of therapeutic objectives is established after the anamnesis stage, according to the evaluated conditions of the patient. Taking these objectives into consideration, sequential, time-limited sessions of exercises that aim to achieve those goals have to be designed by the medical experts. The patients will carry out the rehabilitation sessions, and some intermediate evaluation will be done to check if the patients are progressing and achieving the expected outcome, according to the Goal Attainment Scale (GAS) (Turner-Stokes 2009). In this scenario, physicians need to design combinations of exercises that contribute in a quantitatively measureable way to one or several therapeutic objectives, that might conflict among them, and that might have time, order, intensity or difficulty constraints in order to be selected.

This paper proposes to model the design of rehabilitation therapies by means of Automated Planning, which provides an automatic method to support physicians in the design of these sessions. After their clinical feasibility validation, the therapeutic plan could be projected into a programmable humanoid robotic platform that will serve as training assistant to patients, as expected in the THERAPIST project (Calderita et al. 2013). To achieve this goal, three main steps have been performed and described in this manuscript. Firstly, a domain analysis and specification have been performed with the help of physicians and therapists at Virgen del Rocío Hospital in Seville, as described in the following section. Then, we have studied how to formalize the domain with two different automated planning approaches: classical STRIPS planning and Hierarchical Task Network (HTN) Planning (Ghallab, Nau, and Traverso 2004; Erol, Hendler, and Nau 1994). Finally, we have performed an initial empirical, qualitative evaluation of such models with different planners to check their capabilities, including an extended discussion to highlight their strengths and weaknesses.



[1] http://www.huvr.es

## Domain Analysis and Specification

There are three main actors involved in the therapeutic protocol: the therapist, the medical expert and the patient. The therapy plan is composed of sessions, each one composed of different exercises. The medical expert determines the number of sessions of the therapy and some constraints to prevent the training of certain groups of exercises depending on the patient profile. This expert also decides which general therapeutic objectives, out of five, should be trained during a rehabilitation therapy: bimanual, fine unimanual, coarse unimanual activities, arm positioning or hand positioning activities.

The main goal of the therapist is to help the patient to perform the rehabilitation sessions while evaluating the patient evolution. When planning a therapy session, the therapist selects the exercises which, according to his experience, are better to fulfill the therapeutic objectives in a fixed amount of time. The hospital that participates in our project follows general guidelines of the available exercises categorized according to affected body sites, hence each group trains a certain aspect of a therapeutic objective. The therapist is also free to use his creativity to improvise new exercises in order to better accomplish the goals imposed by the medical expert. A reasonable session might be organized as follows: the initial exercises serve as warming up, the most intense exercises are performed in the central part of the session and the final exercises as cooling down.

The evolution of the patient is evaluated using the GAS scale (Turner-Stokes 2009). Depending on the results, the medical expert can change some therapy features, for example removing a therapeutic objective or adding another one. This system has little flexibility because it does not allow to reduce or increase the priority of the objectives by some degree. The selected exercises in a session depend greatly in the intuition of the therapist. These exercises could not be the most appropriate to achieve the therapeutic objectives and, at the end of the therapy, some of these objectives could not be completely fulfilled, putting at risk the rehabilitation success.

Having different exercises for each therapeutic objective is convenient because using an assorted exercise set may enrich the therapy quality. However, selecting the adequate exercises for the therapeutic objectives, observing the patient profile constraints and the general characteristics of each therapy session, is a difficult and time-consuming search task. For these reasons, the sessions are usually repetitive, using just a small set of performed exercises. This may cause a reduction of the patient's engagement in the therapy.

## Model, Constraints and Requirements

Finding a plan of exercises for each session while taking into account all the requirements set by the medical expert is a difficult search problem that can be solved with Automated Planning. A database with exercises of different characteristics is available for the system to be developed. This database provides metrics to guarantee that the planned therapy fulfills all the requirements of the medical expert. To increase the flexibility when selecting exercises, the terapeutic objectives variables are graded with four adequacy values, $\{0,1,2,3\}$, as used in the GAS scale. These values will contribute to reach the *therapeutic objectives cumulative levels* (TOCL) established for a session. The system will provide the planned sessions to the therapist, that only need to validate or change any exercise considered as not appropriate.

In the following we show the constraints and requirements followed to plan the exercises that will be included in the rehabilitation sessions.

**Goals**

- Total number of sessions.
- Minimum and maximum duration of each session.
- A target level for each therapeutic objective.

**Exercise characteristics**

- Duration (in minutes).
- Adequacy level for each therapeutic objective.
- Intensity value associated to the average hearth rate while performing the exercise.
- Difficulty for a certain patient to perform the exercise. This variable could be updated by the therapist after each session, if needed.
- Each exercise belongs to a group of exercises. These groups are related to the capabilities that patients need to perform the exercise, possibly restricted by their clinical conditions.

In order to preserve the medical pursued requirements and sessions' variability, the next constraints are considered.

**Basic constraints**

- Each session must have three phases in the following order: warm-up, training and cool-down.
- The duration of each phase and each session must be inside a predefined range.

**Variability constraints**

- The repetition of a certain exercise in the same session is not allowed.
- The exercise distribution should be assorted throughout the sessions.

**Patient-related constraints**

- Avoiding a certain group of exercises or a certain exercise (e.g. too much intense or difficult) could be required because of patient conditions.
- Select certain types of exercises (e.g. if the patient suffers "Upper Erb OBPP", recommend only exercises for shoulder abduction, external rotation of shoulder and elbow flexion; if he suffers "Extended Erb OBPP", add wrist flexion as well.
- Within a session, limit the cumulative intensity or difficulty to a given value.

With this information, the automated planner can find a suitable therapy plan if there are enough exercises in the database. In case that the available exercises are not enough, the automated planner will ask the therapist that it needs learning a new exercise with a suggested value for some characteristics. For example, in a session plan, the planner

can suggest the execution (and learning) of a new exercise with a minimum adequation level of 2 for bimanual activities. The planner assumes that the learnt exercise is performed by the patient and uses the minimum estimated values to compute the calculations for the plan. When the therapist saves the new exercise in the database, it can have higher adequation levels for the therapeutic objectives, guaranteeing that the plan will continue being valid. In future sessions, the previously learnt exercises can be reused, minimizing the need of further learning actions and helping the therapist to fill the database with a set of useful exercises. After a session, the therapist can update the difficulty values of the exercises for a patient, if needed. The medical expert can also modify the goals with the results of the GAS scale evaluation. This updates can cause a replanning of the remaining sessions, if the previously planned therapy is no longer valid.

## Methods

We propose the use of Automated Planning techniques (Ghallab, Nau, and Traverso 2004) to plan the exercises that will belong to each session. Automated Planning is an Artificial Intelligence (AI) technique that is used to find a plan of actions while respecting the model constraints. We have tested two different paradigms: classical planning and Hierarchical Task Network (HTN) planning. In classical planning, given a model composed of a initial state, possible actions that have preconditions that need to be fulfilled and effects over the state, and a set of goals that have to be accomplished in the final state, a planner is able to generate valid plans of actions to achieve the goals specified. In HTN Planning (Erol, Hendler, and Nau 1994; Nau et al. 2003) a hierarchy of composed tasks and primitive actions exist. Composed tasks are high-level tasks that can be decomposed using methods that have to fulfill a precondition to be selected and applied, while primitive actions are modeled as in classical planning.

In order to check the viability and to measure the appropriateness and performance of each automated planning paradigm, two different knowledge engineers of our group addressed the presented problem using two concrete AI planners: CBP (García-Olaya, Jiménez, and Linares López 2011) for the classical paradigm and JSHOP2 (Nau et al. 2003) for the hierarchical one. Subsequent meetings with other group experts were carried out to discuss modelling approaches and results. We describe next these two models.

## Classical Planning

The proposed domain for this planning model is based mainly in fluents and action costs, introduced in PDDL 2.1 (Fox and Long 2003). These requirements have yet a lack of support from many of the most current planners, but in this domain they are specially useful to operate directly with the quantitative values of the therapeutic objectives. Also, they are useful to control the session length, add specific variability restrictions and to establish a dynamic preference of certain actions. The most important design criterion that we followed in the classical model is that each individual session has to fulfill always the therapeutic objectives while observing basic time constrains. A secondary criterion consists in forcing the variability among the therapy sessions to avoid monotony and get a much more integral training. This domain also has the possibility to "plan the learning" of new exercises with some suggested attributes to be executed in a session if there are not enough exercises in the database. This learning mechanism is explained in more detail in a later subsection. To clarify the further explanation, below we show a plan for just one session. A full therapy plan will have every session planned, addressing all the dependencies among them.

```
 0: (SESSION-START)
 1: (WARMUP-PHASE)
 2: (WARMUP-DATABASE-EXERCISE E0)
 3: (TRAINING-PHASE)
 4: (TRAINING-DATABASE-EXERCISE E11)
 5: (TRAINING-DATABASE-EXERCISE E12)
 6: (TRAINING-DATABASE-EXERCISE E10)
 7: (TRAINING-DATABASE-EXERCISE E9)
 8: (LEARN-TRAINING-EXERCISE O_SPATIAL_HAND A_MEDIUM
      D_LONG I_INTENSE)
 9: (COOLDOWN-PHASE)
10: (COOLDOWN-DATABASE-EXERCISE E15)
11: (SESSION-END)
```

### Planning Problem

**Goals** The medical expert is in charge of determining the characteristics of the therapy. Firstly, he decides the total number of sessions and the minimum and maximum durations of each phase. This data is stored in the initialization part of the PDDL problem to serve as a common background for all sessions. There are other two important tasks for this expert: choose restrictions depending on the patient's profile and determine the amount of training for each therapeutic objective in each session. These are done using PDDL goals.

There is a fluent for each therapeutic objective to accumulate all the corresponding adequacy values of the planned exercises in a session. These can be defined as the amount of training for a certain therapeutic objective (the aforementioned TOCL values). An objective will be achieved if it is trained enough, so it is sufficient to assign a goal with a lower threshold for each objective to be trained. As an example, for a 30 minutes session:

```
(>= (TOCL t_bimanual) 15)
(>= (TOCL t_unimanual_fine) 7)
(>= (TOCL t_spatial_arm) 7)
```

Numeric goals in PDDL permit a great flexibility to configure the range of desired values for each fluent at the end of each session. It could be also possible to establish an upper limit for each objective (less or equal condition) or even avoid the training of a certain objective (equals to zero), but this has less medical sense because these values are just a form to represent the priority of the therapeutic objectives and do not need to be directly related with the exercises intensity, for which there is a different fluent.

In a similar way, the exercise intensity and difficulty have been modelled with fluents and can be limited for each individual exercise or using a cumulative value for each session. A goal with a standard STRIPS predicate can be used to avoid a certain exercise group.

**Exercise Database** The database contains all the stored exercises. It is fully managed by the therapist, adding exercises when the system suggests it with learning actions or

when the therapist finds it convenient. This information observes all the characteristics of the exercises mentioned in previous sections. The difficulty value of each exercise is stored in the patient's profile, but to simplify we consider that they are loaded in the PDDL problem file before the planning task. To assure the variability constraints, there are two additional fluents representing the session number and the position of the exercise in the the last session where it appeared. Each exercise has a predicate to be able to appear in the warm-up, training or cool-down phase.

The system assumes that the information of the database is coherent, so the therapist has to be sure that the exercises are correct when he adds them. For example, warm-up exercises should not be too intense. With these considerations, the session plan will start with soft intensity, followed by an intense training phase and ending with softer exercises again. Below there is an example exercise modelled in PDDL.

```
(e_phase e7 p_training)
(e_group e7 g_arm_independence)
(= (e_last_session e7) 2)
(= (e_last_position e7) 4)
(= (e_intensity e7) 48)
(= (e_difficulty e7) 39)
(= (e_duration e7) 4)
(= (e_adequacy e7 t_bimanual) 0)
(= (e_adequacy e7 t_unimanual_fine) 3)
(= (e_adequacy e7 t_unimanual_coarse) 0)
(= (e_adequacy e7 t_spatial_arm) 1)
(= (e_adequacy e7 t_spatial_hand) 0)
```

## Planning Domain

**Actions** All actions are strongly based in fluents, having numeric preconditions and action costs. There are two standard action types: to control the session flow and to add exercises.

Flow control actions allow moving among warming up, training and cooling down phases or determining the start and end of a session. If the minimum time for a phase has been reached, it is possible to move to the next phase or to finish the session.

The basic way to add exercises to a session is through actions which select them from the database. They check that there is available time in the current phase and constraints like the maximum cumulative intensity. Only exercises for the current phase can be selected. To assure variability there are two hardcoded restrictions:

- The exercise could not be used in the last three sessions.
- In the training phase, an exercise cannot be trained in the same position as in the last session in which it appeared.[2]

**Learning new exercises**

Learning actions helps the therapist to add new useful exercises to the database as the system is being used. When the planner has difficulties to find a valid plan, it can ask the therapist to teach him a new exercise which execution allows to continue planning. Our hypothesis states that the bigger the database is, the less new learnings will be needed. It is preferable to use exercises in the database instead of learning new ones, but is not necessary to explore all the possible combinations before trying a learning action. This has been controlled using a higher action cost, as a soft goal. The planner tries to minimize the total cost of the plan, so these actions tend to be used few times. The planned learning action has several parameters, as can be seen in the example plan of a session. When one of these actions is used, the planner selects a value for each parameter, having a direct effect on the action cost. The first two (main therapeutic objective and its minimum adequation level) guide the learning to accomplish an unreached goal. The other two parameters are duration and intensity. It is important to highlight that these parameters values are just estimations. The therapist has to be sure that the adequation value of the therapeutic objective pointed by the planner is over the minimum. This way, the therapy plan will continue being valid for the therapeutic objectives. The therapist also has to take into account the suggested duration and intensity, but light variations are acceptable.

To increase variability, the action cost will be higher when the exercise allows to reach the problem goals faster. In other words, exercises longer and less adequate for the main target are preferred.

## Planning Strategy

Classical planning has to deal with a major problem in this domain. Plan only one session is somewhat relatively easy, but a real therapy is composed of about 20 sessions. The first approximation was to plan the full therapy, generating plans which contain more than 250 actions. Planning multiple sessions in one run causes a non-linear complexity increase because there are dependencies among them. The planner has to do backtracking if the selected exercises for a session are not valid. The problem appears when the planner goes back further than needed, maybe many valid sessions, forcing to replan these sessions again to find a valid alternative for a later one. A smaller backtracking of just a few actions could solve the situation allowing reordering of the exercises, selecting others or planning the learning of a new one to continue onward.

Our divide and conquer strategy consists in planning each session individually taking into account the dependencies of one another. In particular, the planner is called one time for each session that we want to plan. Each time that the planner returns a plan, it is parsed to determine all the database and learnt exercises planned. For the next session, a new problem file is generated to update the predicates and functions of the exercises of the last session, and add the new learnt exercises to the database. Then, the planner is executed again with the problem file for the next session. So for each session, a PDDL problem file is generated with the new exercises and updates to the database exercises. The experiments showed that this strategy is much faster than planning all the sessions in one run, without affecting the quality of the plans. The capacity to learn new exercises gives to the sessions some locality properties that can be exploited to avoid the time-consuming backtracking among sessions.

[2]For warm-up and cool-down phases the condition is not applicable because long exercises can exist reaching by themselves the minimum time of the phase and cannot be reordered (in warm-up phase, they will always appear in the first position).

| Sessions \ Planned exercises → | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|---|
| 0 | e0 | e9 | e11 | e12 | e10 | e7 | e15 | |
| 1 | e4 | e2 | e5 | e6 | L | L | L | e13 |
| 2 | e1 | e3 | e8 | L | L | L | L | e16 |
| 3 | L | L | L | L | L | L | e17 | |
| 4 | e0 | e11 | e12 | e10 | e9 | L | e15 | |
| 5 | e4 | e2 | e6 | L19 | e7 | e5 | L20 | e13 |
| 6 | e1 | e3 | L24 | e8 | L23 | L22 | e16 | |
| 7 | L25 | L26 | L30 | L27 | L28 | L29 | e17 | |
| 8 | e0 | e12 | e10 | L31 | e11 | e9 | e15 | |
| 9 | e4 | e2 | L19 | L20 | e6 | e7 | e13 | |
| 10 | e1 | e3 | L22 | L23 | L24 | e8 | e16 | |
| 11 | L | L25 | L26 | L29 | L30 | L27 | L28 | e17 |
| 12 | e0 | e10 | L21 | e12 | e9 | e11 | e15 | |
| 13 | e4 | e2 | e7 | e6 | L20 | L19 | e13 | |
| 14 | e1 | e3 | L24 | e8 | L22 | L23 | e16 | |
| 15 | L25 | L27 | L31 | L26 | e5 | L29 | L30 | e17 |
| 16 | e0 | e11 | e12 | L28 | e10 | e9 | e15 | |
| 17 | L32 | e4 | e2 | L19 | e6 | L20 | e7 | e13 |
| 18 | e1 | e3 | L22 | L23 | L24 | e8 | e16 | |
| 19 | L25 | L26 | e5 | L29 | L30 | e17 | | |

Table 1: Therapy plan with few exercises in the database. An "e" or "L" with a number represents an exercise stored in the database (initial or learnt, respectively). A single "L" represents the learning and execution of a new exercise.

## Empirical Evaluation

We used the CBP automated planner (García-Olaya, Jiménez, and Linares López 2011) because it was specially designed to work with action costs, so its heuristics reduce the total number of new learnings. In Table 1 there is an example of a therapy with 20 sessions. The database starts with a controlled set of exercises: 5 warm-up, 8 training and 5 cool-down exercises.

In session 0, the planner only uses exercises from the database because they are useful to reach the TOCL thresholds. In sessions 1 and 2, it needs to learn due to variability constraints. In session 3, almost all the exercises has been used in the last three sessions, so it has to continue learning new ones. In session 4, the planner can use the set of exercises of session 0 again, but it varies the order of the training phase because the exercises cannot appear in the same position as the last time. In the following sessions, the learnt exercises are reused because they continue being useful to fulfill the goals, so more learning actions are not needed. Note that the sessions are very different among them.

The new learnings in session 4 and 11 show that learning actions are not completely prohibited, so it is not needed to explore all the combinations in the database before using a learning action. Also, we only use the first plan returned by CBP. This planner can improve the plans iteratively if it has time, but the first plan returned by CBP is good enough to see how the system works. With this configuration, the planning time usually does not take more than five minutes. In the initial experimentation, we observed that the principal aspects that increase planning time are the number of learnings needed and the TOCL thresholds.

## HTN Planning

The automatic generation of therapies is a problem that can be managed in a hierarchical way, where the top of the pyramid contain a task representing the whole therapy, which is divided into sessions and each session comprises a set of exercises, as shown in Figure 1. The session structure is given by the hierarchical and order relationships represented in the HTN decomposition. This approach aims to provide a model more easily extendible and configurable, where human expert knowledge can be included at any time.

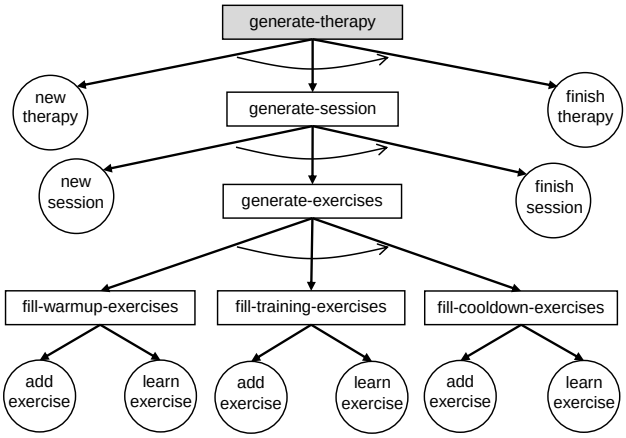


Figure 1: Hierarchical Task Network model schema

## Planning Problem

**Goals** As shown in Figure 1, the goal of the Hierarchical Task Network is the root level of the tree (generate-therapy). This general task comprise three arguments: number of sessions to plan, duration interval for each session and patient identifier. This task can be refined using the HTN decomposition methods until a set of primitive actions complete the plan, which should reach the TOCLs. These TOCLs are also modelled as numeric predicates in the problem description. Furthermore, with the aim to parametrize the search (time and possible exercises in a phase or session), a set of predicates is also included.

**Exercise Database** The exercise database has been modelled similarly to how it has been described for the classical planning approach. The only difference is purely technical due to the representation language of the HTN planner used for evaluation, described later.

## Planning Domain

The HTN planning domain is organized as shown in Figure 1, where a set of exercises is generated for a number of sessions. This behaviour is modelled as a recursive task (generate-session) which receives the current session number and the total number of sessions as parameters. This current session number (?csn) is used as identifier by the planning domain and increased to generate new sessions.

```
(:method (generate-session ?csn ?tsn)
  ;main
  ((call <= ?csn ?tsn))
      ((!new-session ?csn ?tsn)
      (generate-exercises ?csn)
      (generate-session (call + ?csn 1) ?tsn))
  ;stop
  ((call > ?csn ?tsn))
      nil
 )
```

Each session is divided into three phases modelled as lower-level tasks: warm-up phase, training phase and cool-down phase. The system must distinguish which exercises are appropriate for each phase depending on its features, and decides if learning a new exercise is required during planning time.

**Axioms** Axioms allow to infer new predicates from the evaluation of a logical expression (abductive inference). We have defined axioms to control the phases time intervals and to control which exercises are more appropriate for that phase according to the parameters specified in the planning problem. For example, (cooldown-time) and (cooldown-exercise) in Figure 2 are calls to axioms.

Each session begins and ends with less intense and difficult exercises, followed in the training phase with greater intensity and difficulty exercises. The expectation for each session (comprising three phases) is that the values of intensity and difficulty follow a Gaussian distribution, as shown later in the empirical evaluation (see Figure 3). Using axioms throughout the planning domain simplify modelling this requirement.

**Tasks and Methods** Methods are used to refine compound tasks into lower-level tasks or primitive actions. These methods have a precondition that needs to be fulfilled in order to be applied. In our model, we have use five tasks:

1. (generate-therapy) has a unique method with empty precondition that use a total-order decomposition to call the lower-level task (generate-session).
2. (generate-session) is modelled as a recursive task that has a method to call lower-level task (generate-exercises) and a "nil" method that stops when the number of sessions required is reached.
3. (generate-exercises) has a unique method with empty precondition that calls a total-ordered sequence of lower-level tasks (one for each phase).
4. (fill-*phase*-exercises) are modelled with three methods. The first one checks a) that the current time is within the phase time interval, b) that the exercise is suitable for the phase and c) that the exercise selected has not been already included in the ongoing generated session plan. Figure 2 shows a high-level description of how the (fill-cooldown-exercises) task has been modelled. The first method uses a "sort-by" function that drives the planner in the order in which the variable bindings will be evaluated for the method precondition. This "sort-by" function calculates an heuristic value (?ht), modelled as follows:

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

where $d_i$, for each therapeutic objective $i$, is the distance (a minus operation) between the current cumulative level (if the exercise would be included) to the desired TOCL for the planned session. So, the function rewards exercises whose contribution minimize the distance to the frontier solution. The second fraction penalizes the number of times an exercise has been used previously.

The second method is applied when all the possible exercises have been already included in a session, so there is no available exercises to add. In this case a new exercise needs to be acquired (learn action) from the physician.

Exercises will be added taking into account the heuristic and recursive calls to (fill-cooldown-exercises) will be carried till the preconditions fails. In this last case, the third method precondition is evaluated (TOCL reached within the maximum session time specified); if it is fulfilled, the plan is valid, otherwise the planner will do backtracking to check other exercise sets, until this condition is reached.

**Primitive actions** We use dummy actions to delimit start and end of sessions and therapy (see Figure 1). The action to add an exercise updates the current session time (adding the exercise duration) and the current cumulative level for the therapeutic objectives in that session. It also updates the status of the exercise (to "used") and the counter of times used. At the time of writing this paper, the "learn" action establishes fixed values for the exercise attributes. Improving this behaviour is subject of future work.

```
;; Receives the session number                                   Task definition
(:method (fill-cooldown-exercises ?csn)
Method 1
  (:sort-by ?ht >                                               Precondition 1
  ((e-target1 ?e ?et1)
  (current-acc t1 ?csn ?ct1a)
  (baseline t1 ?t1bl)
   ...
  (assign ?d1 (call - ?t1bl (call + ?et1 ?ct1a)))
   ...
  (assign ?h1 (call / 1 (call + (call * ?d1 ?d1) 1)))
   ...
  (e-used ?e ?n-used) (t-session-number ?tsn)
  (assign ?ht (call - (call + ?h1 ... ?h5) (call / ?n-used ?tsn)))
   ...
  (cooldown-time ?cst ?minST ?maxST)
  (cooldown-exercise ?e ?minST ?maxST)
  (not (used ?e ?csn))))
          ((!add-ex ?e cool-down)                               Actions and task calls
          (fill-cooldown-exercises ?csn))
Method 2
(forall (?e) ((exercise ?e)) (used ?e ?csn))                    Precondition 2
          ((!learn))                                            Actions and task calls
Method 3
  ((current-session-time ?csn ?cst)                             Precondition 3
  (session-max-time ?csn ?maxST)
  (call <= ?cst ?maxST)
  (current-acc t1 ?csn ?ct1a) (TOCL t1 ?t1bl) (call >= ?ct1a ?t1bl)
   ...
          ((!finish-session ?csn)))                             Actions and task calls
```

Figure 2: JSHOP2 code for the task that aims to include a set of exercises in the cool-down phase.

## Planning Strategy

Our hypothesis states that a well modelled hierarchical representation of the domain knowledge, along with parameters to drive the search appropriately, could generate successful solutions with a improved quality. In other words, we look

for a parametrized design to provide a more flexible configuration to the physicians. Moreover, in order to reflect the medical criteria in the resulting plan, the heuristic function is in charge of the exercises' selection. As explained before, this function is also used to penalize the most repetitive exercises reducing the heuristic value, but this does not avoid the occurrence of the same exercise throughout sessions. That is why we consider the variability constraints as soft constraints.

The HTN approach can search towards reaching general therapeutic objectives that imply interactions among sessions. These interactions can occur due to a) the exercise distribution of previous ongoing planned sessions that could affect to future ones, b) the TOCL of subsequent sessions can be updated by the plan of earlier sessions and c) chasing possible distributions (eg. time, intensity, TOCLs) for the whole therapy predefined by physicians. This is the motivation to propose a recursive model in order to generate multiple sessions. The HTN approach preserves the capability of backtracking through past sessions without mediation of a external program.

## Empirical Evaluation

We have used the SHOP2 HTN language (Nau et al. 2003) for modelling the planning domain and JSHOP2[3] to test the plan generation. The SHOP2 language is provided with a great expressiveness that allows axiomatic inference, symbolic and numerical computation, call to external programs and use of conditional quantifiers, to name a few of its features.

In order to evaluate the behaviour of the hierarchical domain, a set of 72 exercises are included in the planning problem. This experiment has been carried out with the following configuration: 30 sessions to generate, 25-30 minutes per session, 20% of the total session time is assigned to each warm-up and cool-down phases and the remainder 60% is for training phase. The established intervals to consider an exercise as a candidate for each phase are: warm-up intensity [0-30], warm-up difficulty [0-20], training intensity [30-50], training difficulty [30-50], cool-down intensity [0-20] and cool-down difficulty [0-30]. It is assumed that an exercise could be considered as warm-up and cool-down according to their values. The effects of the exercise distribution is shown in the Figure 3, where the desired Gaussian distribution of the intensity and difficulty with respect the phases is achieved.

## Related Work

There has been some work in the automatic generation of therapy plans or treatments. Ahmed et al. (Ahmed et al. 2010) present a system for the automatic generation of treatments in cancer patients. The system is concerned with the correct selection of the geometry and intensity of the irradiation to produce the best dose distribution. In (Morignot et al. 2010) authors use also automated planning for generating scenarios helping handicapped people. In (Fdez-Olivares et al. 2011; González-Ferrer et al. 2013) authors used a planning algorithm capable of generating oncology treatment plans, and transforming Asbru computer-interpretable guidelines of the Hodgkin disease protocol, which include challenging temporal constraints, difficult to schedule manually by physicians. Schimmelpfeng et al. (Schimmelpfeng, Helber, and Kasper 2012) present a mixed-integer linear programming (MILPs) approach to determine appointments for patients of rehab hospitals. However, they do not plan the specific exercises within each session to achieve some predetermined goals, as detailed in our work.

[3]it uses a planning compilation technique to synthesize domain-dependent planners from SHOP2 domain descriptions

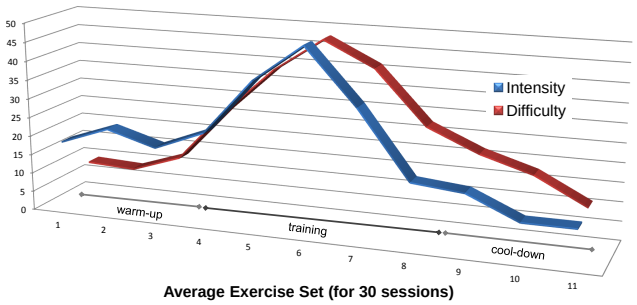


Figure 3: Average results for the HTN model show the desired intensity/difficulty Gaussian distribution.

## Discussion and Conclusions

To conclude this manuscript we have created a qualitative comparison of the two approaches that highlight the main topics addressed. Table 2 represents a summary of this comparison. Some further comments are described next. Firstly, with regard to how to achieve variability while trying to fulfil our requirements, we noted that the sort-by function used in JSHOP2 needs to arrange and order many bindings before the task decomposition is applied, which may affect performance. This could possibly be improved by modelling the heuristic function using a java comparator function, as offered by JSHOP2. Secondly, we think that more flexibility for the user seems to be available in the HTN model, where more complex expert knowledge could be represented easily, however costs and preferences modelling could be very beneficial in the case of using classical planning. Third, using the divide-and-conquer strategy in the classical approach, new learnt exercises can be included in the following sessions, but using the HTN approach learnt exercises can be immediately included into the exercise set. Fourth, the Divide-and-Conquer strategy used in the classical approach eliminates the capability to do backtracking to previous sessions, but improves its time performance. On the other hand, the HTN model can search towards reaching general therapeutic objectives that imply interactions among sessions. Finally, the classical planner CBP does not use heuristics for fluents, so in this domain the search process is very blind. In the case of HTN, it would be interesting to explore new preference-based planning approaches.

| Constraints, Requirements | Classical Planning | HTN Planning |
|---|---|---|
| Assuring Variability | • Avoid repeating exercises in the same session.<br>• Avoid repeating an exercise in the last 3 sessions.<br>• In the training phase, an exercise cannot be trained in the same position than in the last occurrence. | • Avoid repeating exercises in the same session<br>• Heuristic sort-by function penalized for exercises that occur repeatedly |
| Phase Selection | PDDL predicate relating exercise to phase. | axioms limiting the exercise selection whose minimum and maximum duration, intensity and difficulty can be defined by physicians in each phase |
| Phase Time Intervals | Minimum and maximum duration for each phase, manually assigned by the medical expert. | Time is parametrized through axioms according to accumulated percentage of total time for each phase (eg. 0.2, 0.7, 1.0) |
| Learning new exercise | Suggest which attribute values should have a new learnt exercise. | It does not suggest the minimum values yet, but it is already modelled as a new HTN method that can add new exercises during planning time |
| Achieving Goals | • It is a Planner task to achieve the numeric goals established in the planning problem, which are the TOCL thresholds, while observing the constraints.<br>• Includes a metric to minimize the total cost of the plan, where learning a new exercise has more cost than use one from the database. | • Total-order hierarchical network expressing the three phases. Baseline levels and expected session time should be reached, otherwise backtracking occurs to find a suitable exercise set<br>• Driving exercise selection through a sort-by function (see description above) |
| Planning Multiple Sessions | Divide and conquer strategy which calls the planner one time per session. | HTN Planning is done as usual in one run, doing backtracking when exercise sets dont reach expected goals |

Table 2: Qualitative comparison of Classical and HTN approaches for the presented problem

## References


Ahmed, S.; Gozbasi, O.; Savelsbergh, M. W. P.; Crocker, I.; Fox, T.; and Schreibmann, E. 2010. An automated intensity-modulated radiation therapy planning system. *INFORMS Journal on Computing* 22(4):568–583.

Calderita, L.; Bustos, P.; Mejas, C. S.; Fernndez, F.; and Bandera, A. 2013. Therapist: Towards an autonomous socially interactive robot for motor and neurorehabilitation therapies for children.

Erol, K.; Hendler, J.; and Nau, D. S. 1994. HTN planning: Complexity and expressivity. In *AAAI*, volume 94, 1123–1128.

Fdez-Olivares, J.; Castillo, L. A.; Cózar, J. A.; and García-Pérez, Ó. 2011. Supporting clinical processes and decisions by hierarchical planning and scheduling. *Computational Intelligence* 27(1):103–122.

Fox, M., and Long, D. 2003. PDDL2.1: An extension to PDDL for expressing temporal planning domains. *J. Artif. Intell. Res.(JAIR)* 20:61–124.

García-Olaya, Á.; Jiménez, S.; and Linares López, C. 2011. The 2011 international planning competition.

Ghallab, M.; Nau, D.; and Traverso, P. 2004. *Automated planning: theory & practice*. Elsevier.

González-Ferrer, A.; Ten Teije, A.; Fdez-Olivares, J.; and Milian, K. 2013. Automated generation of patient-tailored electronic care pathways by translating computer-interpretable guidelines into hierarchical task networks. *Artificial Intelligence in Medicine* 57(2):91–109.

Morignot, P.; Soury, M.; Leroux, C.; Vorobieva, H.; and Hède, P. 2010. Generating scenarios for a mobile robot with an arm: Case study: Assistance for handicapped persons. In *Proceedings of 11th International Conference on Control Automation Robotics & Vision (ICARCV)*, 976–981. IEEE.

Nau, D. S.; Au, T.-C.; Ilghami, O.; Kuter, U.; Murdock, J. W.; Wu, D.; and Yaman, F. 2003. SHOP2: An HTN planning system. *J. Artif. Intell. Res.(JAIR)* 20:379–404.

Peleg, M. 2013. Computer-interpretable clinical guidelines: A methodological review. *Journal of Biomedical Informatics* 46(4):744–763.

Schimmelpfeng, K.; Helber, S.; and Kasper, S. 2012. Decision support for rehabilitation hospital scheduling. *OR spectrum* 34(2):461–489.

Turner-Stokes, L. 2009. Goal attainment scaling (GAS) in rehabilitation: a practical guide. *Clinical rehabilitation* 23(4):362–70.