A Repair-Replanning Strategy for HTN-based Therapy Planning Systems*

Inmaculada Sánchez-Garzón[1], Juan Fdez-Olivares[1] and Luis Castillo[2]

[1] Department of Computer Science and A.I., University of Granada, Spain
`{isanchez,faro}@decsai.ugr.es`
[2] IActive Intelligent Solutions, Spain
`L.Castillo@iactive.es`

Abstract. This work presents a strategy to respond to failures during the execution of a treatment plan in the real world. In the first place, the approach (focused on HTN-based therapy planning frameworks) performs an exception analysis episode to identify both the nature and the complexity of the unexpected event. According to these factors, the strategy, firstly, will attempt to carry out a *repair* of the failed plan by a local adaptation or applying a suitable repair rule in order to make the plan valid to the current context. However, this alternative might not be effective enough for solving errors that require more global and deeper changes. In this case, the strategy will try to construct another suitable sub-plan through a *replanning* episode for replacing the damaged actions. A *mixed-initiative* approach is also considered, as last alternative, for repairing the error. Aspects related to *plan stability* and *exploitation* of the previous *reasoning process* are also taken into account.

1 Introduction

Generally speaking, the first aim of therapy planning systems is to support the effort of healthcare professionals when they deal with the problem of designing a suitable treatment plan for a given patient. Regarding this topic, Hierarchical Task Network (HTN, [1]) planning systems have proven to be successful on several real therapy planning applications ([2], [3]). The reasons are, on the one hand, that this planning paradigm supports clinical protocols representation, in the planning domain, as a hierarchy of tasks networks. On the other hand, it supports the efficient generation of personalized therapy plans (representing a course of actions to be accomplished at a given time) in order to treat a disease for a concrete patient. In addition, such automatically-generated plans can be seen as *personalized workflows* that may be interactively executed and supervised by healthcare professionals. Hence the second point to achieve in this context is to apply these personalized workflows to *real* patients in a *real* environment. However, the high dynamism of the clinical environment makes long-term therapy plans have to be frequently adapted according to the health conditions and

---

*This work has been partially supported by the Andalusian Regional Ministry of Innovation under project P08-TIC-3572

the progress of the patient. Furthermore, HTN-based approaches construct plans relying on the typical assumptions in classical planning (supposing that everything is going to happen as planned since the world is *static* and *deterministic*) and these suppositions are not realistic in the clinical world.

In order to face this situation, we consider necessary to incorporate flexible adaptation capabilities into the current therapy planning systems for responding to unexpected events happened during the execution of a personalized workflow in the real world. For this reason, our immediate research goal is to develop a generic strategy for tackling any exceptions during the execution of a therapy plan in the dynamic and changing healthcare environment. In addition, such proposal could improve both the acceptance, the diffusion and the effectiveness of the current therapy planning systems at covering a vital part of the life-cycle of plans in clinical domains, i.e., the topic of clinical exception management [4].

Taking the characteristics of healthcare environments into account, we consider that such strategy should fulfill the following desirable features. In the first place, it should support some level of interaction with healthcare professionals to take advantage from their expertise and intuition. This fact requires appropriate communication approaches both for advising about the detected exception and for requesting additional information to repair the plan. On the other hand, since most of the unexpected events are related to changes in *temporal data* (defining the parameters of the patient, as his weight or his height) and these exceptions (not being predicted in advance) are commonly solved with local modifications in the plan, the strategy should be flexible enough for handling both simple and more complex failures. Moreover, it should minimize the changes induced in the therapy plan, thus promoting the *plan stability* [5]. The reason of this conservative attitude is that, when the execution of a therapy plan fails, many of its parts remain unchanged, others have already been executed and another parts may require resources (humans or materials) that have already been committed, thus redirecting a new plan can be costly. In addition, this stability proposition could reduce the cognitive load of healthcare professionals since the suggested plan should be close to the original one that they previously validated before carrying out it in the real world. Finally, the strategy should exploit the reasoning and decision-making process performed by the planner in the construction of the original plan (avoiding as far as possible the replanning *from scratch*). The reason is that planning domains (representing clinical protocols) and therapy plans (representing personalized workflows) are relatively complex (a lot of knowledge entities and restrictions), so a replanning *from scratch* is not recommended to repair a simple and single error.

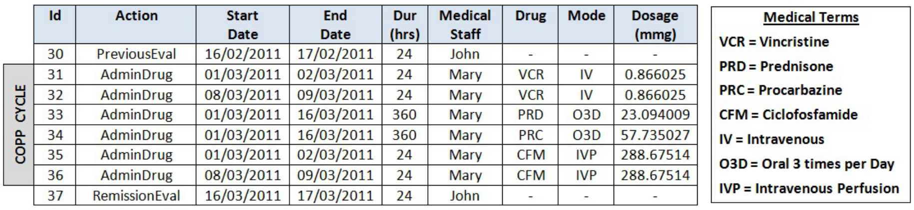
All these requirements, with the poor flexibility of exception handling found on the existing therapy planning system, have motivated the development of a *repair and replanning* strategy as the main topic of my doctoral thesis. The proposed strategy is focused on HTN planning frameworks and it is based on the approach known as *execution monitoring and replanning* [6]. In general terms, the idea of this strategy is to construct the therapy plan in a deterministic form, i.e., without modeling all posible execution exceptions (as conditional branches) on the planning domain. Besides being complicated to predict all posible failures that may happen in the clinical environment, this fact could greatly complicate the protocol modeling. After the generation of the plan, and a previous validation phase of it, the personalized workflow is carried out in the real world and healthcare professionals interact with a *monitoring component* in order to follow the execution trace of the plan and manage the execution of its actions. When an unanticipated event is detected, the system will invoke the repair and replanning (R&R) strategy for solving it.

In the following section, we will introduce the research project in which we participate and the clinical domain that will be considered as case study for the future experimental evaluation of the proposal. After that, and as preliminary experimentation, we will describe the functionality of a monitoring component developed, which is useful from the exception identification point of view. Then, we will present our proposal of repair and replanning strategy that, for the time being, is in its early stage of development. Finally, we will review related work and will conclude with some expected results and open issues.

2 Case Study: OncoTheraper

The work presented in this paper is being carried out in the framework of a research project aimed at developing a clinical decision support system (called *OncoTheraper*) in the paediatrics oncology area. OncoTheraper, which is based on the HTN planning approach [3], is intended to support the effort of oncologists when they deal with the problem of designing an oncology treatment for a given patient. Therefore, planning domains represent protocols from the oncology area in this therapy system. Regarding the topic of exception managing, this kind of protocols is representative for both producing long-term plans and offering a significant number of exceptional situations. In addition, such plans contain collections of actions (with complex temporal restrictions) that may be executed in sequential or parallel flows. In general terms, these actions represent organizational, drug administration and evaluation activities to be accomplished on a specific patient (see figure 1). The second ones (grouped by chemotherapy cycles) are defined by the specific drug, the mode of administration and the dosage to be administer to the patient and this quantity depends on his profile (weight, height, etc.). Evaluation actions are periodically carried out by oncologists in order to follow the progress of the patient.

For the experimentation process of the proposal here presented, we are going to consider the protocol to deal the *Hodgkin's Lymphoma*, which has previously been modeling by our research team and it allows us to generate personalized plans to treat this disease ([7]).



Id	Action	Start Date	End Date	Dur (hrs)	Medical Staff	Drug	Mode	Dosage (mmg)
30	PreviousEval	16/02/2011	17/02/2011	24	John	-	-	-
31	AdminDrug	01/03/2011	02/03/2011	24	Mary	VCR	IV	0.866025
32	AdminDrug	08/03/2011	09/03/2011	24	Mary	VCR	IV	0.866025
33	AdminDrug	01/03/2011	16/03/2011	360	Mary	PRD	O3D	23.094009
34	AdminDrug	01/03/2011	16/03/2011	360	Mary	PRC	O3D	57.735027
35	AdminDrug	01/03/2011	02/03/2011	24	Mary	CFM	IVP	288.67514
36	AdminDrug	08/03/2011	09/03/2011	24	Mary	CFM	IVP	288.67514
37	RemissionEval	16/03/2011	17/03/2011	24	John	-	-	-

Medical Terms

VCR = Vincristine
PRD = Prednisone
PRC = Procarbazine
CFM = Ciclofosfamide
IV = Intravenous
O3D = Oral 3 times per Day
IVP = Intravenous Perfusion

Fig. 1. A simplified treatment following Hodgkin's Protocol

3 Monitoring Component for Detecting Exceptions

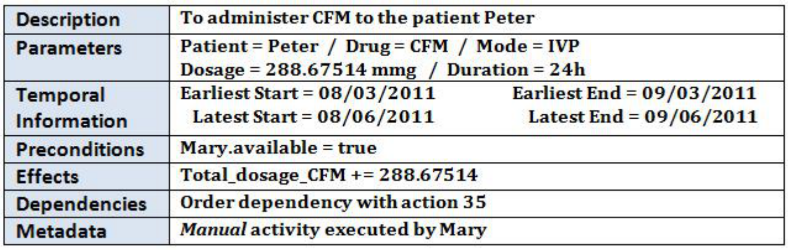
The first step before initiating a repair episode is to detect an exception while the treatment plan is being executed in the real world. An exception occurs when something is not going as planned, i.e., it is any deviation from the steps defined in the original personalized workflow. To support both the execution and monitoring activities, treatment plans are not only represented as a collection of actions. In addition to the description and the parameters, every action contains a set of information entities (as preconditions, effects, temporal information, dependencies and a set of metadata) that are useful from the monitoring standpoint (see Figure 2). Taking these aspects into account, the functionality of the execution and monitoring component developed can be summarized as:

a) Checking that the list of *preconditions* is met in the real world (before executing an activity) and supervising that the list of *effects* of an action has been correctly applied (after its execution).

b) Confirming that activities have initiated and finished their execution according to their *temporal information*. This entity represents the duration and the estimated time for the start and end of each activity. Indeed, the plan is deployed over a temporal constraint network, [8], and these temporal points represent flexible time intervals with the *earliest* and *latest* start and end dates at which an action is allowed to be executed.

c) Plans also have a collection of *order* and *causal dependencies*, which are generated by the reasoning process of the planner. Through these entities, the monitoring component is able to know the correct sequence in which actions must be executed. Furthermore, it allows to keep the execution trace of the plan and to represent each action according to its execution state (initial, ready, executing, finished, suspended, canceled or aborted). On the other hand, a periodical check on the causal structure of the plan is carried out by the monitoring component in order to detect failures as soon as possible, thus having additional time to suggest a valid repair.

d) Finally, a set of *metadata* is used for specifying additional knowledge in actions, as the information entity that differences between *manual* and *automatic* activities (depending on whether they require or not human intervention to be initiated).



Description	**To administer CFM to the patient Peter**
Parameters	**Patient = Peter / Drug = CFM / Mode = IVP** **Dosage = 288.67514 mmg / Duration = 24h**
Temporal Information	**Earliest Start = 08/03/2011 Earliest End = 09/03/2011** **Latest Start = 08/06/2011 Latest End = 09/06/2011**
Preconditions	**Mary.available = true**
Effects	**Total_dosage_CFM += 288.67514**
Dependencies	**Order dependency with action 35**
Metadata	***Manual* activity executed by Mary**

Fig. 2. Information entities of the activity 36 (AdminDrug)

Regarding the extract of the plan shown in figure 1, the preceding illustration shows the information entities defining the administration action 36. In this example, the field of temporal information shows the time interval in which this action may be carried out. This fact means that, if the treatment plan started five months later, it would remain valid since a correct temporal propagation process would be carried out in order to guarantee the temporal patterns defined in the oncology protocol. On the other hand, the precondition expresses that *Mary* (who is the nurse in charge of administering the drug to the patient *Peter*) must be available. After an administration action, the total quantity of a specific drug that has been applied to the patient is updated and this fact is defined by the effects of the action. In addition, this activity has an order dependency with action 35, i.e., for executing action 36 it is necessary that activity 35 be finished. Finally, this action is marked as *finished* in the execution component by *Mary* since it is a *manual* activity and she is in charge of performing it.

Since these previous entities are explicitly represented in plans, the monitoring component can autonomously detect errors related to them. However, there are exceptions identified in runtime by healthcare professionals that such component must be able to support, e.g., the patient suffers a fever or his progression is not as planned. This kind of unexpected events also requires to be handled so, after detecting an error, the repair and replanning strategy will be invoked.

4 A Proposal of Repair-Replanning Strategy

After the detection of a failure, an episode of exception analysis is carried out in order to identify both the failed activity, the cause of such error and the list of actions affected by it (called *impact* of the failure). It is possible to know such list through an analysis of the *causal structure* of the plan, which is generated by the reasoning process of the planner. If the unexpected event has been identified by the monitoring component (in an automatic way), a notification of such contingency will be sent to the healthcare professionals. However, if the exception has been detected by such experts, they must report on the failure to the system and offer it some feedback information to guide the adaptation process. This process is carried out by the repair and replanning (R&R) strategy that we propose. Its design and its structure have been motivated by the desirable features that, from our point of view, a generic and effective repair mechanism should fulfill. As a result, the R&R strategy is organized into different levels (see figure 3) according to the criticality and the nature of the exception detected.

Hence the two first levels are aimed at carrying out *repairs*, i.e., applying local changes or a *repair rule* to adapt the treatment plan. Nonetheless, the objective of the third level (which produces more global changes in the plan) is to initiate a *replanning* episode for generating a new sub-plan according to the current context. Finally, a *mixed-initiative* approach is also considered (as last alternative) in the fourth level. Other aspect to comment is that the degree of autonomy of the proposal and the degree of user involvement depend on the complexity of the failure to solve, i.e., the resolution of complex errors (e.g., the patient has not progressed as planned) requires more expert participation than solving a simple one (e.g., a doctor is not available). This fact also involves that the resolution of simpler errors may be carried out in a semi-automatic way.

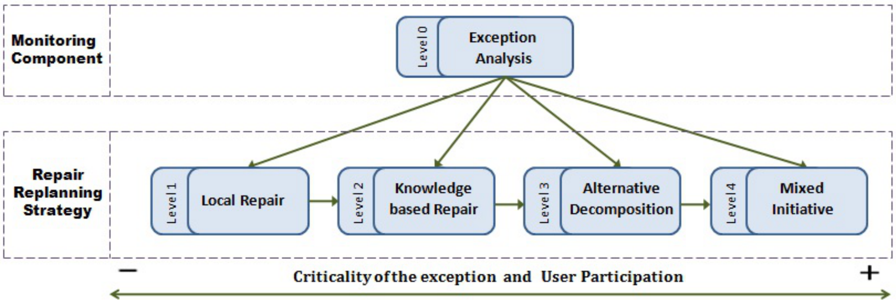


Fig. 3. Overview of the Repair-Replanning Strategy

In order to avoid the replanning *from scratch*, the proposal relies on the previous decision-making process carried out by the planner for the construction of the original plan. For this reason, the sequence of decisions made by the planner during the plan construction are recorded in an additional structure called **decision graph**. Taking the HTN paradigm into account [1], the nodes of this structure represent the planning domain entities through which the reasoning process goes, such as compound tasks (goals), decomposition methods (the alternatives to achieve a goal) or primitive tasks (actions). An arc between two nodes represents the order restriction of these HTN entities in the planning domain. Each node of this graph (action, method or goal) records the decisions performed by the planner (in planning time) in the corresponding HTN entity and we identify the following decisions: (1) the list of *resources* whose attributes made true the preconditions of a primitive task or of a decomposition method (list of *unifications*) and (2) the method selected to decompose a compound task. In the case of nodes related to goals, the decision graph also records the list of alternatives (other decomposition methods) for reducing the compound task and some information related to the internal state of the planner, which is very useful to initiate a replanning episode, as seen below.

Figure 4 shows an extract of the decision graph related to the treatment plan of figure 1. In such illustration, the action node *RemEval* has recorded a decision related to the specialist (the oncologist *John*) who, in runtime, will perform the activity to the patient. Nevertheless, the goal node *EvalRem* has been decomposed by the method *CompRem* and the treatment plan finishes (at considering that the tumour will have a complete remission). However, such goal has alternative decomposition methods according to the posible health conditions of the patient after the chemotherapy cycles. These different methods specify the set of actions that are necessary to perform if the tumour has a partial remission, or if the disease of the patient is stable or progressive, respectively. All these actions are defined according to the specification of the oncology protocol.

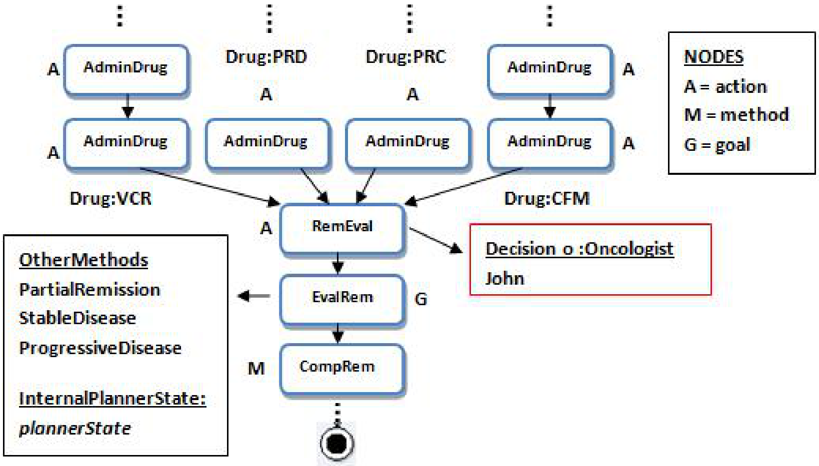


Fig. 4. An extract of the decision graph

In the next sections, we explain briefly the functionality of each level of the R&R strategy. The election of the level for adapting the plan is made in the step of exception analysis according to the nature of the detected error.

4.1 Level 1: Local Repair

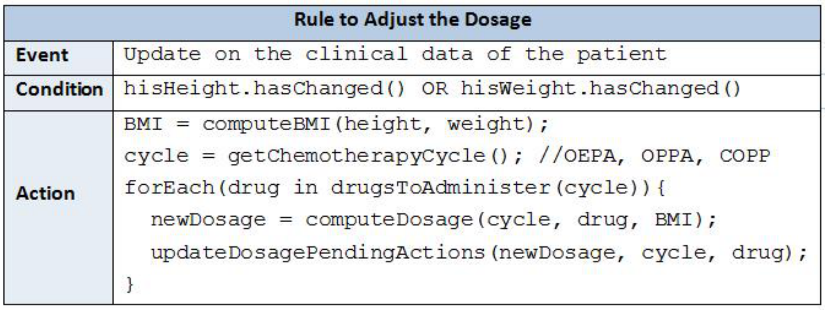
Our planner manages information related to resources (human or material) and, during the generation of the plan, it allocates them to the activities of the workflow. We consider that a large number of exceptions are caused by errors in resources that make the preconditions of an activity fail. For this reason, the repair in this level is based on finding a suitable replacement of the failed resource that makes these preconditions be satisfied, thus allowing to carry on with the execution of the plan. A simple example of this kind of error is that the oncologist allocated to an evaluation session is not available in runtime (since the irregular work timetables of healthcare professionals) and the solution is to find a suitable candidate for performing the activity. Regarding resources, we mention a set of assumptions that must be considered. In some cases, the selected resource may affect the duration of the activity in which is used. For example, the duration of an evaluation session might depend on the surgery times of the selected oncologist or the duration of an administration activity might depend on the administration time of the specific drug selected. In other cases, the resource may be selected to a sorting criterion, which is useful for establishing priorities among resources. For example, choose the doctor with the best availability. Therefore, these assumptions must be taken into account to suggest a close candidate resource.

The repair in this level is summarized in these steps: (1) Get the list of candidate resources with the same type as the failed one (depending on if it is an oncologist, or a nurse, or a material resource, etc.). (2) If the failed resource was selected by a shorting criterion, the list of candidate resources will also be sorted before checking it. (3) Analyze the list until we find a compatible resource that satisfies the preconditions of the failed action, mets its temporal restrictions (if the resource affects the duration of the activity) and it does not cause any *threats*. This latter point involves to analyze the causal structure of the plan and the other activities in which this candidate resource is used and check that, sure enough, there is not overlap of the resource (not used simultaneously in several actions). (4) If a compatible resource has been found, the plan is automatically updated with this new candidate.

4.2 Level 2: Knowledge-based Repair

Although an effective R&R mechanism should be a domain-independent strategy, we are aware that in the clinical context the type of repair to apply depends on the considered application domain, on the detected error, on the current health conditions of the patient and even on the experience of clinicians on previous episodes. Moreover, there is some kind of *expected* exceptions that, although it is difficult to exactly predict them in advance, we know that they are going to happen during the execution of a treatment plan (due to the dynamic nature of this environment). For example, in the case study considered in this work, the body mass index (BMI) of the patient (which is a temporal data depending on his height and his weight) affects on the dosage of the administration actions. Since plans are generated following a long-term approach, such quantity must be adjusted when the body mass of the patient has changed. We are aware that this kind of exceptions is going to happen, but we do not know exactly *when* and *in which* measure this temporal data will vary.

Therefore, the objective of this level is to provide a collection of pre-defined *repair rules* in order to solve these common exceptions in an automatic way. Such repair rules are based on the ECA rules ([9]), which are defined by three components: *event* (the signal that triggers the invocation of the rule), *condition* (the logical test that, if it is evaluated to true, causes the action to be carried out) and *action* (the set of steps to adapt the treatment plan). Figure 5 shows an intuitive example of an ECA rule, which is in charge of adjusting the dosage of the administration actions of a treatment plan. In this case, when the monitoring component detects that the weight or the height of the patient have changed (after an updating process of his clinical data), an activity to adjust the dosage is carried out. In the first place, the new value of his BMI is computed. Secondly, it is necessary to know what chemotherapy cycle is being applied to the patient since this aspect affects both on the type of drugs and the dosage to administer. After that, we compute the new dosage for each drug according to the previously calculated parameters and we update the pending actions of the plan with this value. This updating process involves to change some parameters of the actions (its *dosage* parameter) and even the effects that use this value. For instance, the *effects* field in figure 2 increments the total quantity of a specific drug that has been administered to the patient, so this field also requires an adaptation.



Rule to Adjust the Dosage

Event	`Update on the clinical data of the patient`
Condition	`hisHeight.hasChanged() OR hisWeight.hasChanged()`
Action	

```
BMI = computeBMI(height, weight);
cycle = getChemotherapyCycle(); //OEPA, OPPA, COPP
forEach(drug in drugsToAdminister(cycle)){
  newDosage = computeDosage(cycle, drug, BMI);
  updateDosagePendingActions(newDosage, cycle, drug);
}
```

Fig. 5. Pseudocode of an ECA rule

The main objective of this level is to have a complete set of pre-defined repair rules to cover the typical *expected* exceptions that may occur during the execution of a treatment plan. Some examples of these rules are: a *temporal delaying rule*, that postpones the pending actions of a treatment plan until a desirable state is satisfied (e.g., to delay the plan until the fever that suffers the patient has remitted). This delay is made by a *temporal constraint propagation* process. Another rules may be a *canceling rule*, to remove a specific action of the plan (e.g., a high toxicity level in the patient forces the oncologist to delete an administration action) or an *adding rule*, to incorporate a new action in the treatment plan (e.g., in order to protect the patient against infections after a bad reaction of a drug, the oncologist adds an action to administer antibiotics). Repair rules related to *reorder* the actions of a treatment plan or *replace* some parameters of them (e.g., change this drug, for this one) will also be considered.

4.3 Level 3: Alternative Decomposition

The previous levels seen so far try to make local changes for solving the failed plan. However, if this proposal is not effective enough for complex errors, the solution here is to initiate a replanning episode for replacing the damaged actions. The main motivation of this level is that long-term oncology plans are generated by an optimistic attitude, i.e., the tumour will have a complete remission after applying all the chemotherapy cycles to the patient. But, unfortunately, this philosophy is not realistic in clinical environments. Hence the objective of this level is to achieve the goal (to treat the disease of the patient) in a different way (applying another decomposition method) according to the current context (the real health conditions of the patient). In this case, the plan might require deeper modifications (such as including or deleting actions or replanning a complete chemotherapy cycle), which are achieved with a new planning episode.

When an exception is detected, the steps to carry out in this level are the following: (1) identifying (in the decision graph) the first high-level node related to a compound task t to which the next invalid action a of the plan belongs; (2) checking that none of the activities that belong to the task network of t have initiated their execution (in the case that exists a parallel flow of activities); (3) checking if the goal t has other decomposition methods m; (4) confirming that the preconditions for applying the method m are satisfied in the current context and, if so, (5) invoking the planner from t with m and taking the internal planner state recorded in t into account. If the planner does not return a valid sub-plan, we try with other decomposition method of t. When all the alternatives have been proved, the following solution is to *go up* in the decision graph until a more high-level node t' and try to replan from it, if t' satisfies the previous steps. If the planner returns a valid sub-plan, this one will be merged into the original plan and the new fragment of the decision graph generated will be incorporated into the original one. For instance, taking the decision graph shown in figure 4 into account and regarding that the patient has a partial remission of the tumour instead of a complete remission; an example of this proposal is to invoke the planner from the goal *EvalRem*, with the decomposition method *PartialRemission* and with the corresponding internal state of the planner (*plannerState*).

Finally, if the exception analysis process of the level zero considers that the unexpected event can not be solved by none of the previous levels seen so far or, conversely, none of them has suggested a valid repair; the solution here is to invoke the mixed-initiative approach of the fourth level of the R&R strategy.

4.4 Level 4: Mixed-Initiative

The aim of this level is to propose a framework in which healthcare professionals can interact to the R&R strategy in order to guide the plan adaptation process. The objective is to take advantage from their intuition and their expertise for solving the detected error. In general terms, the previous levels are related to *expected* exceptions that commonly are identified and solved by the monitoring component in a semi-automatic way. However, the idea of this last level is to support both the failures identified by healthcare professionals in runtime and the decisions made by such experts through an interactive framework that allow them to adjust, organize, delete or add activities in the treatment plan in order to make it valid to the current context. In this case, and regarding the detected exception, clinicians might adopt two different approaches. On the one hand, they might add some actions to the failed plan in order to achieve a state (i.e., a consistent state) in which a set of conditions are met and this state allows to invoke the planner as in the third level. On the other hand, they could add actions that might constitute a new conditional branch of the clinical protocol. Moreover, clinicians might add both actions that are pre-defined in the planning domain and completely new actions (specifying, in this case, their parameters, preconditions and effects). Finally, verification mechanisms are also required for guaranteeing, somehow, the correctness and consistency of the repaired plan suggested by the experts.

Once explained the R&R strategy, we are going to review related work and we will conclude this paper with some expected results and open issues.

5 Related Work

The idea of adapting a failed therapy plan has been discussed from different perspectives, as in [10], which presents a generic approach for handling unexpected events in runtime. Such unexpected events are classified into *hazards* and *obstacles* exceptions, which are simpler errors. A limitation of this work is that healthcare professionals can not dispose of the *complete* treatment plan that will be applied to a patient because the list of goals to achieve are *refined* during plan execution. This fact means that actions, which are planned for a very short-term horizon, are added to the plan *on-demand*, i.e., almost when they are ready to be executed. For this reason, this approach makes difficult to estimate, in advance, both the set of actions and resources that will be required in the future and it might even complicate their later commitment.

Regarding the area of Artificial Intelligence Planning and Scheduling (AI P&S), we consider that most plan repair approaches are focused on non - hierarchical planning, as [5] and [11]. Other works, such as [12], present a hybrid approach that integrates HTN decomposition. However, this approach enables the generation of abstract plans that include parts that will be refined in later stages. From our point of view, this work presents the same limitation as [10] since clinicians can not count on the complete treatment plan to be applied to a patient and, as is well known in oncology, specific drugs for a chemotherapy cycle must be ordered, in advance, before the planned administration actions.

On the other hand, taking the properly HTN-based paradigm into account, we mention several works. In [13], the repair approach consists of re-executing the actions that make true the failed preconditions, but this approach neither is effective enough nor realistic for clinical contexts. Other related works are [14], [15] and [16], where the repair strategies rely on additional structures as our *decision graph*. However, we consider that these proposals do not offer a flexible response for tackling exceptions in the healthcare environment since they try to initiate a replanning episode in any case (both for simple and complex errors). In addition, if there is not any appropriate decomposition method to begin the replanning, the strategy will not offer any solution.

Another related area is that of workflow systems, where issues concerned to exception handling have been investigated. [4] presents a brief review related to exception managing in the medical workflow systems. Furthermore, most of the existing works in this area are focused on rule-based approaches, such as, [17], [18], [19] and [20]. However, we consider that this alternative might not be effective enough for exceptions requiring more global changes in the plan since the adaptation process is only based on applying pre-defined rules. Finally, we want mention the work [21], which describes the ADEPT project and proposes some challenges to be considered in order to develop the next generation process management technology. Some of these ideas might be used to describe more exhaustively the fourth level of our R&R strategy, where the interaction with healthcare professionals is a key aspect.

6 Expected Results

This work proposes a repair and replanning strategy (focused on HTN-based therapy planning systems) for tackling unexpected events during the execution of a treatment plan in the real world. Once the exception is detected, either by the monitoring component (in an automatic way) or by the clinicians, an exception analysis episode is carried out in order to know its nature. After that, the R&R strategy, which is organized in four different levels according to the complexity of the exception to solve, is invoked to make the plan valid to the current context. Although the proposal is in an early stage of development, we consider that it has several advantages.

On the one hand, flexibility is a *sine-qua-non* condition of every exception handling approach and we consider that the proposal here presented encourages this principle at covering both local and more global changes, with different degrees of user participation and an enough coverage for solving a wide variety of errors. For this reason, the R&R strategy might be seen as a contribution from the medical standpoint since its incorporation into the current therapy planning systems could improve their effectiveness, acceptance and diffusion. On the other hand, this proposal is also innovative in the area of HTN-based planning as it is a flexible and *quasi*-generic approach that can be applied in domains that are different from the clinical contexts. Moreover, aspects related to *efficiency* (reuse the previous reasoning process of the planner), *stability* (minimize the changes induced in the plan, thus reducing the cognitive load of clinicians) and *interactivity* with experts, are also considered in the development of the strategy.

Regarding the experimental evaluation of the proposal, we can mention that the research project *OncoTheraper* ([2]) is also constituted by a group of qualified oncologists that periodically evaluates the achievements of the developer team.

Such evaluations are based on proofs of concept focused in specific parts of our technology. For example, in order to evaluate the process for generating personalized plans to treat the Hodgkin's disease, a web application has been developed for this purpose ([7]). After fill in a form with the clinical data of the patient, a treatment plan is automatically generated. On the other hand, with the intention of disposing a first prototype of tool for the interactive execution of such plans by clinicians, we have integrated the output of our planner (a treatment plan) with the input of a standard BPM runtime engine (a business process). This integration process ([22]) was carried out through a model-to-model transformation, which can be seen as a contribution in the Knowledge Engineering area at supporting a rapid prototyping development life-cycle. Moreover, this first execution tool might be improved with the functionality of our monitoring component, which is able to detect exceptions autonomously. The objective of presenting this previous experimentation is to demonstrate that the future proofs of concept that will be designed to evaluate the R&R strategy, will be supported by a robust and proven technology. Such proofs allow us both to know the utility of the strategy and to identify new exceptions that must be covered with the collaboration of the oncologists.

7 Open Issues

Since this work is in its early stage of development, there are some aspects that require a more exhaustive study. In this section, we are going to mention some open issues related to the R&R strategy proposed. Regarding the first level of the proposal, and in order to carry out a more effective replacement of resources, it could be useful to consider inter-plan coordination. This topic, which is specially dealt in the area of multi-agent planning [23], involves to have a centralized monitoring component in charge of analyzing all the treatment plans in which each resource (human or material) participates in order to check overlaps or threats more accurately.

On the other hand, taking the pre-defined rules of the second level of the strategy into account, a more formal representation of them is required. The aim is to know more exactly *how* they must be specified (i.e, in which language or nomenclature) and *where* they must be defined (i.e., if they are an additional part of the model representation of the oncology protocol or, conversely, if these repair rules compose a plugin of exception handling of the monitoring component). Another relevant aspect is related to *who* is in charge of defining these rules, the knowledge workers who model the protocol or the healthcare professionals who use the therapy planning system. In this last case, a more intuitive and expressive approach for exception representation must be developed. Finally, a well-argued taxonomy of exceptions might be useful for identifying and classifying unexpected events more properly.

Regarding the third level of the strategy, it is necessary to incorporate into the technology of our planner, a process to automatically generate the decision graph structure which is required for initiating re-planning episodes. On the other hand, the mixed-initiative approach represented in the last level, requires a more exhaustive study to support features as interactivity, usability, flexibility and repair verification in order to propose a framework in which clinicians can define the adaptation process of the failed plan.

The last topic to mention is related to the experimental evaluation of the proposal. In order to prove the utility of the R&R strategy, it could be necessary to model other paediatrics oncology protocols and, even, to consider planning domains from other areas (different from the clinical contexts). This supposition could promote the *quasi*-domain-independent approach of the proposal. To conclude this work, say that our final objective is to create a framework that integrates the required components for covering the complete life-cycle of plan management in the clinical environment.

References

1. E.D. Sacerdoti. The nonlinear nature of plans. In *Proc. of IJCAI*, pages 206–214, 1975.
2. J. Fdez-Olivares, J. A. Cózar, and L. Castillo. OncoTheraper: Clinical decision support for oncology therapy planning based on temporal hierarchical tasks networks. In *Knowledge Management for Health Care Procedures. LNCS*, volume 5626, pages 25–41, 2009.
3. J. Fdez-Olivares, L. Castillo, J. A. Cózar, and O. García-Pérez. Supporting clinical processes and decisions by hierarchical planning and scheduling. *Computational Intelligence*, 27:103–122, 2011.
4. M. Han, T. Thiery, and X. Song. Managing exceptions in the medical workflow systems. In *Proc. of the 28th International Conference on Software Engineering*, pages 741–750, 2006.
5. M. Fox, A. Gerevini, D. Long, and I. Serina. Plan stability: Replanning versus plan repair. In *Proc. of ICAPS*, pages 212–221, 2006.
6. S. Russell and P. Norvig. *Artificial Intelligence: A Modern-Approach.* Prentice Hall, 2009.
7. IActive Technology and OncoTherapy Team. Demonstrator for generating plans following the Hodgkin's Protocol. `http://tratamientos.iactive.es/`.
8. L. Castillo, J. Fdez-Olivares, O. García-Pérez, and F. Palao. Efficiently handling temporal knowledge in an HTN planner. In *Proc. of ICAPS06*, pages 63–72, 2006.
9. N.W. Paton. *Active Rules in Database Systems.* Springer, 1999.
10. A. Grando, M. Peleg, and D. Glasspool. A goal-oriented framework for specifying clinical guidelines and handling medical errors. *Biomedical Informatics*, 43:287–299, 2010.
11. R. van der Krogt and M. de Weerdt. Plan repair as an extension of planning. In *Proc. of ICAPS05*, pages 161–170, 2005.
12. J. Bidot, B. Schattenberg, and S. Biundo. Plan repair in hybrid planning. In *KI 2008: Advances in Artificial Intelligence. LNCS*, volume 5243, pages 169–176, 2008.
13. X. Wang and S. Chien. Replanning using hierarchical task network and operator-based planning. *Recent Advances in AI Planning*, 1348:427–439, 1997.

14. G. Boella and R. Damiano. A replanning algorithm for a reactive agent architecture. *Artificial Intelligence: Methodology, Systems, and Applications*, 2443:359–387, 2002.
15. I. Warfield, C. Hogg, S. Lee-Urban, and H. Muñoz Avila. Adaptation of hierarchical task network plans. In *Proc. of FLAIRS*, pages 429–434, 2007.
16. N.F. Ayan, U. Kuter, F. Yaman, and R.P. Goldman. HOTRiDE: Hierarchical ordered task replanning in dynamic enviroments. In *Proc. of ICAPS*, 2007.
17. U. Greiner, R. Müller, E. Rahm, J. Ramsch, B. Heller, and M. Loeffler. AdaptFlow: Protocol-based medical treatment using adaptative workflows. *Methods of Information in Medicine*, 44:80–88, 2005.
18. Klein M. and C. Dellarocas. A knowledge-based approach to handling exceptions in workflow systems. *Computer Supported Cooperative Work*, 9:399–412, 2000.
19. Greiner U. Müller, R. and E. Rahm. AgentWork: a workflow system supporting rule-based workflow adaptation. *Data and Knowledge Engineering*, 51:223–256, 2004.
20. S. Quaglini, M. Stefanelli, G. Lanzola, V. Caporusso, and S. Panzarasa. Flexible guideline-based patient careflow systems. *Artificial Intelligence in Medicine*, 22:65–80, 2001.
21. P. Dadam and M. Reichert. The ADEPT project: a decade of research and development for robust and flexible process support. Challenges and achievements. *Computer Science. Research and Development.*, 23(2):81–97, 2009.
22. J. Fdez-Olivares, I. Sánchez-Garzón, A. González-Ferrer, J. A. Cózar, A. Fdez-Teijeiro, M. R. Cabello, and L. Castillo. Task network based modeling, dynamic generation and adaptative execution of patient-tailored treatment plans based on Smart Process Management technologies. In *Under revision by the Organization of AIME'11*, 2011.
23. K. P. Sycara. Multiagent systems. *AI Magazine*, 19(2):79–92, 1998.