Planning Models for Two-Way Avoidance and Reversal Learning ⋆

Bernd Schattenberg * **Andreas L. Schulz** ** **André Brechmann** *** **Frank W. Ohl** **** **Susanne Biundo** †

* *Ulm University, Ulm, Germany (e-mail: bernd.schattenberg@uni-ulm.de).*
** *Leibniz Institute for Neurobiology, Magdeburg, Germany (e-mail: andreas.schulz@lin-magdeburg.de)*
*** *Leibniz Institute for Neurobiology, Magdeburg, Germany (e-mail: andre.brechmann@ifn-magdeburg.de)*
**** *Leibniz Institute for Neurobiology, Magdeburg, Germany (e-mail: frank.ohl@lin-magdeburg.de)*
† *Ulm University, Ulm, Germany (e-mail: susanne.biundo@uni-ulm.de).*

Abstract: Reinforcement learning models can explain various aspects of two-way avoidance learning but do not provide a rationale for the relationship found between the dynamics of initial learning and those of reversal learning. Artificial Intelligence planning offers a novel way to conceptualize the learners' cognitive processes by providing an explicit representation of and reasoning about internal processing stages. Our hybrid planning and plan repair approach demonstrates that the empirically found relationships could be motivated from a consistent theoretical framework.

Keywords: Artificial Intelligence, Mathematical Models, Formal Languages

1. INTRODUCTION

The framework of reinforcement learning (RL), in particular temporal difference learning, is traditionally utilized when it comes to modeling Pavlovian and instrumental learning in animal and human subjects. Neurophysiological correlates' model parameters, on the level of both single cells and fMRI data, support this approach. Notably, most of these models are applied to appetitive learning, as their nature demands that reinforcement is directly related to an action. In aversive learning this is not the case – at least not for directly observable actions of the animal. Therefore, RL models of avoidance learning are studied only relatively recently. Previous work (Schulz et al. (2011)) has presented a study to investigate an extended RL model in the context of two-way avoidance and reversal learning experiments that involve Mongolian gerbils trained in a shuttle box Go/NoGo paradigm (Wetzel et al. (2008); Ohl et al. (2001)). The animals learn to discriminate classes of sound stimuli and to respond to them by leaving and staying in their box compartment, respectively. Wrong responses are answered by slightly unpleasant feedback. After some time, the experimenter introduces a contingency reversal: positive and negative feedback criteria are switched. The animals adapt to this contingency but they do so not as efficient as in their initial learning phase. RL models can explain various aspects of two-way avoidance learning but they do not provide a rationale for the relationship found between dynamics of initial learning and dynamics of reversal learning. As we will show in this paper, the Artificial Intelligence (AI) paradigm of automated planning (Nau et al. (2004); Biundo et al. (2011b)) is able to provide such a rationale.

In contrast to RL techniques that aim at an operational level, we use AI planning to conceptualize the animal's cognitive processes as explicit representations of and reasoning about internal processing stages. In a first step, the learning experiment is formalized as an abstract specification of the trial series without contingency reversal. From that the planning system derives a concrete configuration of cognitive processes that is consistent with the specified subject's decision making procedure and with its knowledge about the world: a "best-case" reference plan that describes the animal in non-contingency scenario trials.

Reversal learning can then be expressed by means of reconfiguring the cognitive processes in order to address the changes in the environment. A contingency reversal is consequently interpreted as the occurrence of an execution failure of the reference plan. We use AI plan repair techniques to add knowledge/behavior updates as explicit learning steps, where necessary, or to alter the implementation of complex sub-processes. These techniques are designed to introduce only a minimal amount of change to the previous plan structure: in some parts of the plan, it will be "easier" for the repair procedure to include a knowledge update process that fixes a number of wrong-based decisions of the subject, in other parts, minimal

⋆ This paper reports work done within the Transregional Collaborative Research Centre SFB/TRR 62 "Companion-Technology for Cognitive Technical Systems" funded by the German Research Foundation (DFG).

change translates into merely discovering and assessing that the subject made a wrong decision. Hence, the repair method does not necessarily prevent negative feedback for the subject, but it becomes a qualitative model for the effort it takes to re-establish a consistent cognitive process and with that an estimate for the prediction of subject performance in reversal learning experiments.

2. HYBRID PLANNING

Formal Framework: The hybrid planning framework is a combination of the state-based methods from partial-order causal-link (POCL) and the procedure-oriented techniques found in hierarchical task-network (HTN) planning (Estlin et al. (1997); Kambhampati et al. (1998); Biundo and Schattenberg (2001); Gerevini et al. (2008)). As it is usual for POCL approaches (McAllester and Rosenblitt (1991)), hybrid planning systems conceptualize the application domain in a formal way based on the notion of *states*, which describe all relevant aspects of world situations in terms of object entities and the relationships between them, and by *state transitions*, which account for the dynamics in the world by specifying how situations change.

Our hybrid planning framework makes use of an order-sorted logic and defines states as interpretations of logical formulae over a given language L in which we specify sorts, constants, variables, relation symbols, and the like, that we are going to use in our model. Constants represent typed objects and relations are used to describe properties of and relationships between these objects. For example, a literal or *fact* like $\texttt{detected}(\texttt{stimulus\_1})$ holds in all states where the subject perceived the stimulus `stimulus_1`, and an atom $\texttt{analyzed}(\texttt{stimulus\_1}, \texttt{up})$ is true in every world state in which the subject has evaluated the modulation characteristic of `stimulus_1` and determined its value to be `up`, i.e., it is modulated upwards.

Actions are the means to represent the world dynamics. Their execution induces changes of a situation in the real world world, which means that the interpretation of facts and their respective logical symbols change; the resulting situation is described by a new state that reflects this new interpretation. While the fact $\texttt{detected}(\texttt{stimulus\_1})$ may not hold in a given state, it may do so in a state reached by executing a suitable perception operation in which the subject actively becomes aware of the stimulus. Basic actions are supposed to be executed directly in the environment and are called *primitive tasks*. A primitive task is given by a schema $t(v_1, \ldots, v_n) = \langle pre, eff \rangle$, which specifies the parameters, *precondition* and *effects* of the task t: if the formula *pre* holds in a state, t can be executed in that state and causes a state transition such that formula *eff* holds in the resulting state. For the purpose of this presentation, we may assume that preconditions and effects are conjunctions of positive and negative literals over the logical language L. Further details of state definitions and task semantics can be found in (Biundo and Schattenberg (2001)).

The following task schema specifies an action that corresponds to the process of selecting that feature of a stimulus, which the subject considers to be relevant: $\texttt{select}(f : \texttt{Feature}, s : \texttt{Stimulus}, v : \texttt{Value}) = \langle \texttt{detected}(s) \wedge \texttt{relevantFeature}(f) \wedge \texttt{hasCharacteristic}(s, f, v), \texttt{selected}(f) \rangle$ The precondition of the primitive task requires that the subject has detected a stimulus s and considers a given feature f to be relevant. Furthermore, the characteristics for the given stimulus and feature are determined: note that this is not a state feature intended in the sense of a goal that "has to be made true" but rather in a way of "binding" the parameter v to the respective `Value` object in the current state. After the task `select` has been executed, feature f becomes `selected` for the given stimulus s in the subsequent state.

Before we take a closer look at abstract actions, let us consider the representation of plans. A *partial plan* is a tuple $P = \langle PS, \prec, V, C \rangle$, which consists of a set of *plan steps* (partially instantiated task schema), a set of *ordering constraints* $\prec$ that impose a partial order on the plan steps, a set of *variable constraints*, and a set of *causal links*. The variable constraints are equations for codesignating and non-codesignating parameters that occur in PS with each other and with constants, respectively. Causal links are the means to establish and maintain causal relationships among the tasks in a partial plan: $(s \rightarrow_\phi s')$ means that plan step s provides condition ϕ for plan step s' with $\mathit{eff}(s) \Rightarrow \phi$ and $pre(s') \Rightarrow \phi$.

From HTN planning (Erol et al. (1994)), hybrid planning adopts the concepts of primitive and abstract tasks. By doing so, partial plans do not only represent intermediate results of the plan generation process but are also used to specify *implementations* of abstract tasks. *Abstract tasks* have the same syntactical structure like primitive ones but are associated with at least one *decomposition method* that describes a (standard) solution of the task by providing a partial plan for it. A domain model typically contains more than one method per abstract task and their implementations may in turn include abstract tasks as well. With that, hierarchies of tasks and associated methods can be used to encode the various ways in which an abstract task can be accomplished.

In the discussed domain, we model a trial in the experiment series by the abstract task $\texttt{trial}(s : \texttt{Stimulus}, f : \texttt{Feedback}) = \langle \neg\texttt{presented}(s), \texttt{isGiven}(f) \rangle$. While on the abstract level conducting a trial is specified in terms of giving feedback to the subject on a yet unprocessed stimulus, the partial plan for achieving this task introduces those plan steps that actually present the stimulus to the subject, produce and observe its response, and eventually provide the appropriate feedback to that response.

```
method 1 for trial(s: Stimulus, f: Feedback)
steps
  present(s_p:Stimulus),
  subjectTrialPerformance(s_s:Stimulus, r_s:Response),
  feedback(f_f:Feedback, r_f:Response, s_f:Stimulus)
orderings
  present≺subjectTrialPerformance≺feedback
constraints
  s=s_p=s_s=s_f,   r_s=r_f,   f=f_f
links
 (present →_presented(s) subjectTrialPerformance),
 (subjectTrialPerformance →_responded(s,r_s) feedback)
```

Plan step `feedback` thereby refers to a corresponding abstract task schema. The model provides two methods for that task, one implementing a positive and one implementing a negative feedback from the experimenter, depending on the experimental setup. To this end, the atom `isGiven(feedback)`, which is used in the effects of the abstract task schema definition `trial` above, is specified as an abstract state feature via a so-called *decomposition axiom.* Hybrid planning domain models make use of decomposition axioms in order to allow for causal reasoning between tasks on different levels of abstraction. Regarding the above model excerpt, the axiom $\forall f$: `Feedback`.[`isGiven`(f) $\Leftrightarrow$ `isPositive`(f) $\vee$ `isNegative`(f)] realizes such a link between the abstract action of giving neutral feedback and its concrete implementations.

With the formal structures defined above, we introduce a hybrid planning *domain model* $D = \langle L, \mathcal{T}, \mathcal{M}, \Delta \rangle$ over the logical language L as a set $\mathcal{T}$ of available abstract and primitive task schemata, a set $\mathcal{M}$ of methods for the abstract tasks in $\mathcal{T}$, and a set Δ of decomposition axioms.

A hybrid *planning problem* is given by the 4-tuple $\Pi = \langle D, s_{init}, P_{init}, g \rangle$, with D specifying a domain model, s_{init} an initial state, g a goal state, and P_{init} an initial partial plan. As usual in the POCL paradigm, the initial state specifies the situation supposed to be valid at the beginning of plan execution while the goal describes the requirements for the world state that has to be reached. In addition to that, hybrid planning problems include in an HTN fashion an initial (abstract) plan and require the solution of the planning problem to be a *refinement* of that plan. Applying a decomposition method to an abstract task is an intuitive act of making a plan more concrete, because it replaces the abstract task by one of its predefined solutions. Hybrid planning, however, distinguishes various classes of plan refinements. Besides task expansion they include task insertion, causal link insertion, the insertion of variable bindings, temporal constraints, and the like. Details about plan refinement in this context can be found in (Schattenberg (2009)).

More formally, a partial plan $P = (PS, \prec, V, C)$ is a solution to the hybrid planning problem $\Pi = \langle D, s_{init}, P_{init}, g \rangle$ if and only if the following criteria are met:

- PS contains a plan step *init* that has the initial state s_{init} as effects and a plan step *goal* that has the goal state description g as precondition.
- P contains no abstract plan steps and can be obtained from P_{init} by repeated decomposition of plan steps and insertion of plan steps, causal links, ordering, and variable constraints.
- For every precondition literal ϕ of a plan step $s' \in PS$ there is a causal link $(s \rightarrow_\phi s') \in C$ such that s has ϕ as an effect and s is ordered before s' by the ordering constraints in $\prec$.
- No causal links are threatened, i.e., for each causal link $(s \rightarrow_\phi s') \in C$ the ordering constraints ensure that no plan step $s'' \in PS$ with a literal $\neg\phi$ in the effects can be ordered between s and s'.

These solution criteria ensure that for every linearization of ground instances of the plan steps in PS that are consistent with V, a solution to a hybrid planning problem is executable in the initial state, implements the initial plan, and generates a state satisfying the goal description. Such a solution P ist obtained from Π by *search* in the space of partial plans: Starting in P_{init}, the planning algorithm systematically applies available refinements to the current plan, thereby generating successor plans, which in turn are then subject to refinement application, until a plan is processed that does satisfy the solution criteria (Kambhampati (1997); Nau et al. (2004); Biundo et al. (2011a)). Since an exhaustive search in the plan space for all possible solutions is generally infeasible, dedicated planning strategies focus on promising refinement candidates. They typically sacrifice completeness of the search procedure for the sake of finding "good" solutions and to find them very efficiently, respectively. Their technical details are out of the scope of this presentations, hence we refer the reader to (Schattenberg et al. (2005, 2007)). It is however important to point out that planning strategies can be designed to operate on qualities of the examined plans as well as of the refinements; we will draw on both in the sections below.

Reference Plan Construction: Let us come back to the animal learning experiments: the following fragment of a problem specification defines a small scenario in which a trained subject performs one trial.

```
domainModel
 // sort definitions [...]
 constants
   volume, modulation: Feature,
   high, low, up, down, neutral: Value,
   GO, NOGO: Decision,
   jump, sitNwait: Response,
   stimulus_1: Stimulus,
   feedback_1: Feedback
 // relations, task schemata, methods [...]
initialState
  hasCharacteristic(stimulus_1, modulation, up)∧
  hasCharacteristic(stimulus_1, volume, high)∧
  isHit(stimulus_1, jump)∧relevantFeature(volume)∧
  associatedDecision(volume, high, GO)∧
  associatedDecision(volume, low, NOGO)∧
  associatedDecision(modulation, neutral, NOGO)∧
  associatedDecision(modulation, up, NOGO)∧
  associatedDecision(modulation, down, NOGO)∧
  associatedResponse(GO, jump)∧
  associatedResponse(NOGO, sitNwait)
goalState
  —
initialPlan
  steps
    trial(stimulus_1, feedback_1)
  orderings
  constraints
  links
```

The initial state represents both the experimental setup and the gerbil's internal situation. The modelled stimulus of the given characteristics is supposed to be answered by a jump response. The animal has been trained and we may assume that it regards the volume of a stimulus to be the relevant feature and that it decides to recall the jump behavior on high volume feature values, only. No goal state is specified and hence all partial plans that are executable in the initial state and accomplish the abstract task `trial` under the given circumstances are eligible solutions.

With the decomposition method presented above, the initial plan can be refined into the three steps of stimulus presentation, subject performance and feedback generation.

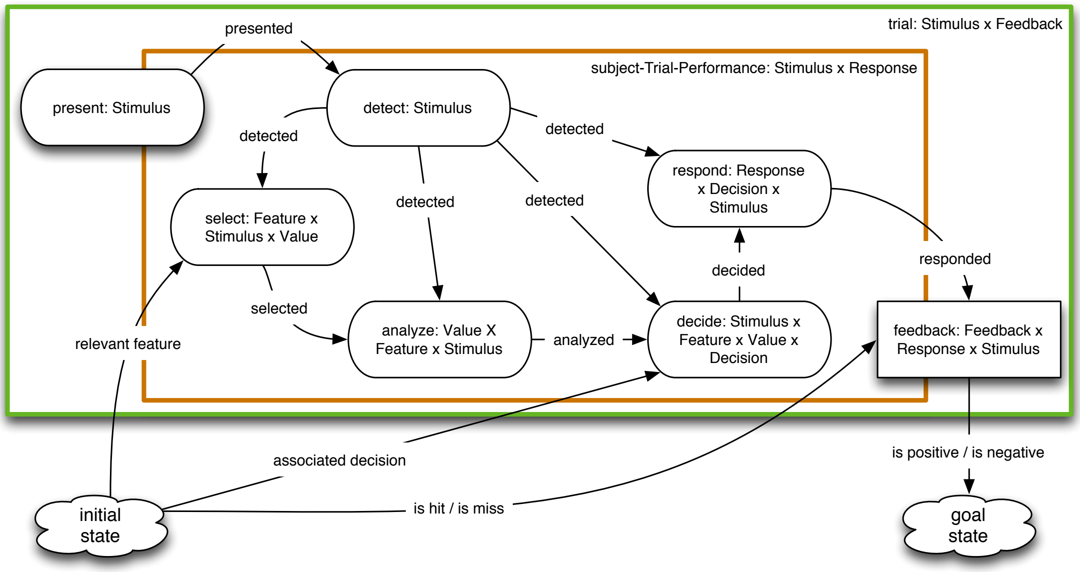

Fig. 1. A simplified implementation of the 1-trial scenario. Boxes depict abstract tasks, arrows causal links.

Fig. 1 outlines the result of applying one more decomposition refinement: the subject performance implementation, shown in the inner box, consists of five primitive tasks that constitute the subject's cognitive process. First, the presented stimulus is detected and its relevant feature determined and evaluated. The arrows denote the most important causal links, which, among others, preserve a consistent treatment of the presented stimulus during the process. In order to formulate a GO/NOGO decision on the detected stimulus, the decision plan step acquires the actual value of the stimulus from the preceding analysis and the subject's association from its memory. The response step also relies on the Gerbil's experience (a corresponding causal link from the initial has been omitted in the figure) when processing the decision and produces as effect the articulation of a response, the state feature $\texttt{responded}(\texttt{jump})$. The feedback step will be decomposed in a subsequent plan refinement and finally compares the response to the evaluation matrix and, in this case, consequently generates a positive reaction.

Please note that the combinatorics in the sketched solution to the 1-trial experiment scenario are solved by exploring the space of plan refinements. The decision task-parameter in plan step `decide` is bound by establishing a causal link from the initial state over the ground literal $\texttt{associatedDecision}(\texttt{volume}, \texttt{high}, \texttt{GO})$. Also, the final feedback implementation depends on whether the subject's response is a hit, a false alarm, or a miss; the initial state defines only neutral feedbacks.

As we did for a single trial scenario, we can now define a whole experiment, which typically comprises 30 to 100 trials. By solving this large planning problem, our hybrid planning system generates what we call *reference plan*, basically a sequence of stimulus presentations, cognitive processes and responses, and feedbacks. It is "a" reference plan, because the planning domain model defines more tasks and methods than we present in this paper, which allows for a variety of alternative implementations of the cognitive processes. E.g., in addition to the presented standard method for the abstract $\texttt{subject} - \texttt{Trial} - \texttt{Performance}$, we provide one with an "inverse" decision task, thus mimicking explorative behavior. As we will see in the repair phase below, the model also contains tasks for manipulating the subject's beliefs, e.g., the planner can insert plan steps that explicitly represent learning by updating the decision association facts if negative feedback has been received. In this way, we can also relax the assumption that the animal is trained prior to the first trial and generate a plan for bootstrapping the cognitive processes in the experimental environment.

It is important to point out that the model per se does neither guarantee nor even prefer successful trials. This is because our task and method portfolio is explicitly intended to allow for false alarm and miss responses, and the plan generation process is only concerned with constructing consistent solutions according to the problem specifications. Generally speaking, the planning approach resembles more an observation of what implication the design of cognitive processes has on the subject's coping with the given experimental situation. At this point, the planning strategy becomes an integral part of the model: in the course of plan space exploration, it monitors the rate of hit responses in each refinement and uses this information to guide the search. The planning strategy, e.g., tolerates a certain amount of negative feedback unless the subject's lack of success exceeds some threshold; it then prefers the application of appropriate exploration implementations or the inclusion of learning steps. In this way, the planning system is capable of constructing reference plans that comply with an expected behavior, either according to an animal's training phase or to predictions of RL models.

3. PLAN REPAIR

Repairing Hybrid Plans: It is a very common assumption in automated planning that the only way in which the environment changes is through the execution of plan steps, exactly in the way they have been designed. In many real-world scenario, this is obviously not always the case and plan-execution may fail for many reasons, ranging from unexpected events to errors in the problem specification. One way to address this problem is by *repairing* the failed plan such that it circumvents the source of the execution problem while at the same time staying as close as possible to the previously obtained course of action. The rationale behind this procedure is that we regard the domain model to remain a correct representation of the world and that whatever caused the current deficiency is an exceptional event. This is also the case for experiencing contingency reversal: the established way of reacting is invalidated, the biological system however has to be convinced that this change is really a permanent one.

Fig. 2 illustrates our approach: Let plan G in the left plan space be a solution to a planning problem's initial Plan A, generated via the depicted refinements. Let furthermore the execution of G fail for some known reason, say, a causal requirement for a plan step is not found satisfied. In the analysis of the planning process, we find, e.g., that plan refinement 3 introduced the eventually faulty causal link in order to obtain a plan D. Plan C is therefore the best developed backtracking point on the way down to solution G, which does not suffer from the execution problem. Syntactic analysis shows that refinement 4 does not depend on the application of 3, whereas 5 builds upon the causal link again. Finally, refinement 6 is not related to 3 and 5 and hence also not affected by the failure.

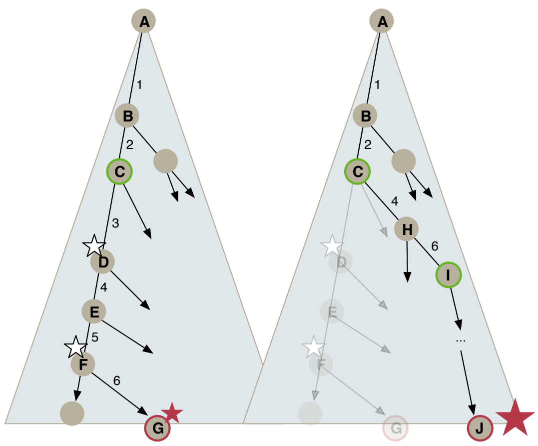

Fig. 2. An example for plan repair: the explored plan space for a solution G (left) and its re-construction for obtaining a repaired solution J (right).

The result of this analysis leads to a reconstructed plan space as shown on the right-hand side: The "safe" plan refinements 4 and 6 are re-applied to the "safe" plan C, resulting in a new and unaffected plan I. This is a good candidate for finally obtaining a new solution J by the same hybrid planning methods that we have introduced in the previous sections. J is also similar to the old solution G in the sense that only small parts of the previous plan have been changed by the repair procedure as far as they have been affected by the execution failure. The search process is thereby guided by the old plan development as reference and at the same time aware of the problems that have been experienced during G's execution (Bidot et al. (2008); Biundo et al. (2011a)).

Contingency Reversal: The reference plan has been constructed on the assumption of stable environmental conditions, i.e., although the subject may receive a certain amount of negative feedback, the interpretation of the `isHit` relation remained constant. We may also assume that either the experimental setup introduces a trained subject or the Gerbil succeeds in correctly learning to discriminate the relevant stimulus feature and in identifying the correct decision/response associations.

We now introduce the contingency reversal, i.e., the definition of hits is inverted and consequently the causal support for some of the concrete feedback plan steps becomes invalid. As shown in the repair procedure above, the system undoes all unappropriate feedback implementations and basically implies three repair options: First, the feedback can be inverted, which will decrease the hit rate of the subject. Second, the implementation of the subject's trial performance can be altered into an exploratory one, which however implies at least undoing one more method decomposition. Third, the system can insert single plan steps for changing the associations of decisions and responses or the relevance of features, which will imply a re-consideration of many causal links in the plan. These briefly described repair options are constantly under consideration by the repair mechanism, while at the same time it develops a consistent solution for the new circumstances that is as similar to the original plan as possible. Fig. 3 illustrates the situation for three trials after coping with a contingency. Let trial B and C be affected by the induced change, trial A is either executed before the contingency reversal takes place or the feedback on the stimulus response remains the same as before. The system undid the implementation of the feedback in trial C to a negative feedback, which is a prerequisite of a newly added explicit learning step $\mathtt{update-decision}$. This new step consumes the old decision association (cf. initial state specification in Sec. 2), together with the negative experience of C, and produces a new state feature that represents the updated knowledge. The implementation of B does not have to be changed, it is only the causal linking of the decision step that has to be re-established - the feedback remains the same.

It is easy to see that the amount of change the repair process induces on the reference plan highly depends on the model of the cognitive processes as well as on the structure of the acquired knowledge about the environment. Both aspects are currently under examination: which processes should be modelled as abstract tasks with implementation alternatives, which are realistic conditions for knowledge update operations. The actual values for repair operations are also under debate: while it is very natural to measure an exchange of implementation in terms of the number of nested (sub-) implementations, there is no clear corresponding notion between the number of re-established causal links and new task insertions.

4. CONCLUSION

The AI planning model has two particularly interesting properties in the context of this work: First, the causal structure of the model explicitly conveys all interdependencies of the subject's acquired knowledge as well as its cognitive processes. Second, the strategy concept allows for

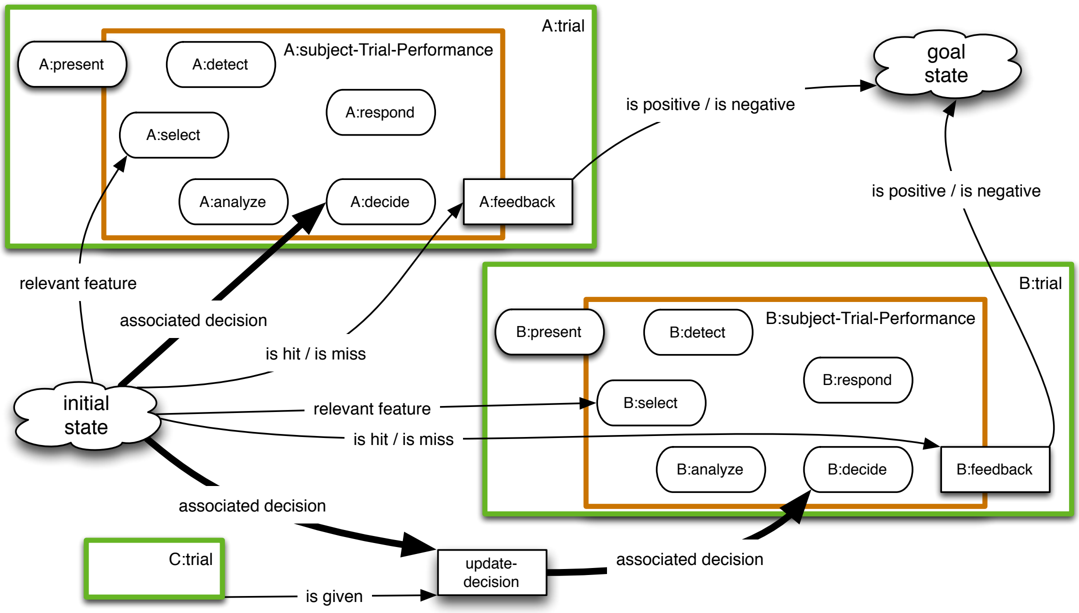

Fig. 3. Three repaired trial implementations after coping with a contingency reversal.

a natural representation of principles like introducing minimal change, avoiding negative feedback, and the like. Our hybrid planning and plan repair approach thereby demonstrates that the empirically found relationships between initial learning and reversal learning could be motivated from a consistent theoretical framework.

It is our ongoing work to quantify the induced change to the reference plan as an adequate estimate for the effort a gerbil has in coping with contingency reversal. The results will be used for experiment design and comparison of cognitive process models. Future work also aims at more integrated models of RL and planning (including probabilistic approaches) and at investigating details of the conceptualization like the complexity of the subject's feature selection mechanism or the granularity of the knowledge update and decision processes.

REFERENCES

Bidot, J., Biundo, S., and Schattenberg, B. (2008). Plan repair in hybrid planning. In *Proc. 31st German Conf. on AI*, volume 5243 of *LNCS*, 169–176. Springer.

Biundo, S., Bercher, P., Geier, T., Müller, F., and Schattenberg, B. (2011a). Advanced user assistance based on AI planning. *Cognitive Systems Research*, 12(3-4), 219–236. Special Issue on Complex Cognition.

Biundo, S., Bidot, J., and Schattenberg, B. (2011b). Planning in the real world. *Informatik-Spektrum*, 34(5), 443–454.

Biundo, S. and Schattenberg, B. (2001). From abstract crisis to concrete relief: A preliminary report on combining state abstraction and HTN planning. In *Proc. 6th European Conf. on Planning*, 157–168.

Erol, K., Hendler, J., and Nau, D.S. (1994). UMCP: A sound and complete procedure for hierarchical task-network planning. In *Proc. 2nd International Conf. on AI Planning Systems*, 249–254. AAAI Press.

Estlin, T.A., Chien, S.A., and Wang, X. (1997). An argument for a hybrid HTN/operator-based approach to planning. In *Proc. 4th European Conf. on Planning*, volume 1348 of *LNCS*, 182–194. Springer.

Gerevini, A., Kuter, U., Nau, D.S., Saetti, A., and Waisbrot, N. (2008). Combining domain-independent planning and HTN planning: The duet planner. In *Proc. 18th European Conf. on AI*, 573–577. IOS Press.

Kambhampati, S. (1997). Refinement planning as a unifying framework for plan synthesis. *AI Magazine*, 18(2), 67–98.

Kambhampati, S., Mali, A., and Srivastava, B. (1998). Hybrid planning for partially hierarchical domains. In *Proc. 15th National Conf. on AI*, 882–888. AAAI Press.

McAllester, D. and Rosenblitt, D. (1991). Systematic nonlinear planning. In *Proc. 9th National Conf. on AI*, Volume 2, 634–639. AAAI Press.

Nau, D.S., Ghallab, M., and Traverso, P. (2004). *Automated Planning: Theory & Practice*. Morgan Kaufmann.

Ohl, F.W., Scheich, H., and Freeman, W.J. (2001). Change in pattern of ongoing cortical activity with auditory category learning. *Nature*, 412, 733–736.

Schattenberg, B. (2009). *Hybrid Planning & Scheduling*. Ph.D. thesis, Ulm University, Germany.

Schattenberg, B., Bidot, J., and Biundo, S. (2007). On the construction and evaluation of flexible plan-refinement strategies. In *Proc. 30th German Conf. on AI*, volume 4667 of *LNCS*, 367–381. Springer.

Schattenberg, B., Weigl, A., and Biundo, S. (2005). Hybrid planning using flexible strategies. In *Proc. 28th German Conf. on AI*, volume 3698 of *LNCS*, 249–263. Springer.

Schulz, A.L., Schattenberg, B., Woldeit, M.L., Brechmann, A., Biundo, S., and Ohl, F.W. (2011). Reinforcement learning and planning models for two-way-avoidance and reversal learning. In *Proc. Annual Meeting of the Society For Neuroscience*. Washington, USA.

Wetzel, W., Ohl, F.W., and Scheich, H. (2008). Global vs. local processing of frequency-modulated tones in gerbils: an animal model of lateralized auditory cortex functions in humans. *Proc. Natl. Acad. Sci. USA*, 105, 6753–5758.