Explaining Failures Propagations in the Execution of Multi-Agent Temporal Plans

Extended Abstract

Gianluca Torta
Università di Torino, Dip. Informatica,
Torino, Italy
gianluca.torta@unito.it

Roberto Micalizio
Università di Torino, Dip. Informatica,
Torino, Italy
roberto.micalizio@unito.it

Samuele Sormano
Università di Torino, Dip. Informatica,
Torino, Italy
samuele.sormano@edu.unito.it

ABSTRACT

This paper addresses the problem of explaining execution failures in Temporal Multiagent Plans (TMAPs). A diagnosis identifies faulty actions (*primary failures*), and those that were affected by fault propagation (*secondary failures*). Temporal explanations group diagnoses for increasing understanding.

KEYWORDS

Temporal Multi-agent Plans; Diagnosis; Explanation; SMT

ACM Reference Format:
Gianluca Torta, Roberto Micalizio, and Samuele Sormano. 2019. Explaining Failures Propagations in the Execution of Multi-Agent Temporal Plans. In *Proc. of the 18th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2019), Montreal, Canada, May 13–17, 2019,* IFAAMAS, 3 pages.

1 INTRODUCTION

Multiagent plans (MAPs) accomplish complex goals by decomposing them into subgoals, and then organizing the activities of a team of agents. Plan execution, however, is not always straightforward. The actual execution of actions, in fact, can be affected by failures. When a failure occurs, detecting and diagnosing it is of primary importance in order to resume the nominal execution.

Some recent works [1, 4, 5] have addressed the problem of diagnosing the execution of a MAP. These works, however, do not model time explicitly, but only implicitly by assuming a sequence of discrete time steps at which atomic actions are performed. As a consequence, anomalous deviations on the duration of actions cannot be detected and diagnosed. Other approaches [6–8], on the other hand, have focused only on the temporal dimension disregarding that a faulty action may not achieve all (or some) of its expected effects. Both families of approaches, thus, can't handle applications where durative actions may take longer than expected, as well as fail in producing the expected results.

This paper addresses the diagnosis of the execution of a Temporal-MAP (TMAP), that is a plan where both missing effects and temporal deviations can occur. We adopt a consistency-based notion of diagnosis: a MAP diagnosis is a subset of actions whose non-nominal behavior is consistent with the observations received so far. The Z3 Satisfiability Modulo Theories (SMT) solver [2], is exploited for inferring all the minimal rank (i.e., most likely) diagnoses consisting in an attribution of a (possibly faulty) modality to each plan action. Diagnoses are then complemented with a set of *temporal* explanations highlighting contingent causal dependencies that might have occurred, and help a user gain the awareness of how a primary action failure has caused other secondary ones.

Proc. of the 18th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2019), N. Agmon, M. E. Taylor, E. Elkind, M. Veloso (eds.), May 13–17, 2019, Montreal, Canada.

2 PLAN EXECUTION FAILURE PROBLEM

A rigorous formalization of a TMAP and of a Plan Execution Failure (PEF) problem can be found in [9], due to lack of space we give here just some intuitive notions. Figure 1 exemplifies a TMAP in the logistic domain. The picture shows the causal (dashed edges) and precedence (solid edges) constraints between actions. It is expected that a nominal execution of the plan will satisfy all these constraints. However, the actual execution could be affected by action failures whose consequences may lead an agent to violate both types of constraints. Indeed, the temporal aspect is modeled by associating each action a with a set of modalities $M(a)$:$\{N, F1, F2, \ldots\}$. The nominal modality N denotes the duration (as an interval) of the action under normal condition, and has rank 0, meaning that is preferred to other, faulty modes. The remaining faulty modes in $M(a)$ are associated with ever greater durations and ranks and with the specification of what effects will be missing when a is performed in that mode. As usual in model-based diagnosis, a PEF problem arises when, given a set of observations Obs, the nominal hypothesis, that is the assumption that each action behaves in its

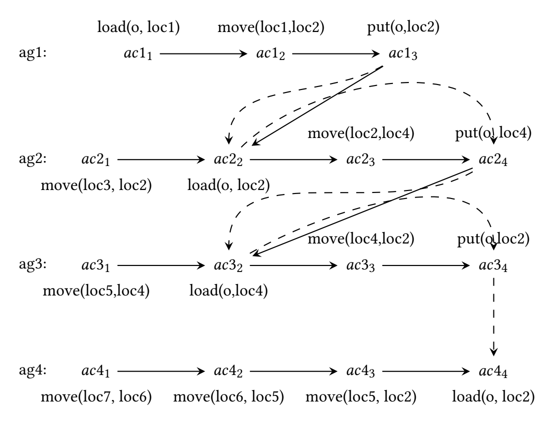

Figure 1: An example TMAP.

normal mode N, is no longer consistent. Note that in our framework observations have the form $\langle e, t\rangle$, where e is a literal (e.g., $at(ag1, p1)$ and $\neg at(ag1, p1)$), and t is the time at which the observation is captured. When the nominal hypothesis is no longer consistent with Obs, a diagnostic inference is needed to find a set Δ of alternative hypotheses that are consistent with Obs. The set Δ is a solution to a PEF problem if each hypothesis $\delta \in \Delta$ is an assignment of modes to every action in the TMAP, such that it is consistent with Obs and its rank (order-of-magnitude inverse probability, see [3]) is minimal. Where the rank of δ is simply defined as the sum of the ranks of the action modes in δ. A solution δ to a PEF problem highlights with a special mode F_P that some actions have been affected by previously occurred action failures. For example (Fig. 1), the $move(loc3, loc2)$ may have a delay (mode $F1$), so that the object released by $ag1$ at $loc2$ is actually loaded by $ag4$. This situation makes actions $ac2_2$, $ac2_4$, $ac3_2$, and $ac3_4$ fail with mode F_P. The diagnosis δ_{ex} (except for N modes) lists: $ac2_1(F1)$, $ac2_2(F_P)$, $ac2_4(F_P)$, $ac3_2(F_P)$, $ac3_4(F_P)$.

3 EXPLAINING FAILURE PROPAGATIONS

A diagnosis as the one just discussed above is not sufficient, for the user, to understand what has actually happened. In general, failures can propagate via the shared literals, that is, via the services produced by an action and consumed by another one. For example, an action may fail because one of the required inputs is not available at the right time, and this may happen because the producer failed in supplying it (including supplying it with too much delay), or because another action has erroneously consumed the service in its place. Explaining δ, thus, means tracing back the temporal relations among the actions that are related to some shared literal of interest, and whose occurrence justifies a secondary failure.

To this end, we define a *Temporal Explanation* $E(\delta, R)$ of δ w.r.t. a shared literal R as a set of Allen algebra relations (i.e., *before*, *after*, *during*, . . .) among the actions in P that produce/consume R. A (full) explanation of a diagnosis δ is simply a set $E(\delta)$ of several sub-explanations $E(\delta, R)$, one for each shared literal [1].

Let us refer to diagnosis δ_{ex} from section 2. The *producers* of literal $R = at(o, loc2)$ are $ac1_3$, and $ac3_4$; while the *consumers* of R are: $ac2_2$, and $ac4_4$. Figure 2 shows $E(\delta_{ex}, R)$ graphically on a diagram where time increases from left to right. Note that, besides the actions related with R and their Allen algebra relations specified by $E(\delta, R)$ (black), the schema also shows some other actions with mode assignments specified by the diagnosis δ (gray); such actions are depicted just to further increase the information conveyed by the schema to the reader.

The set of non-F_P actions that have to do with R are just $ac1_3$ and $ac4_4$, so that the timeline is partitioned in five regions (dotted vertical bars): before $ac1_3$; during $ac1_3$; between $ac1_3$ and $ac4_4$; during $ac4_4$; after $ac4_4$. The definition of explanation requires us to relate $ac1_3$ and $ac4_4$, and a possible scenario is $ac1_3$ *before* $ac4_4$, i.e., when $ac1_3$ ends, some time passes before $ac4_4$ becomes ready and consumes $at(o, loc2)$. Note that R would be available for other consumers between the end of $ac1_3$ to the start of $ac4_4$. However, according to explanation $E(\delta_{ex}, R)$, $ac2_2$ becomes ready and then fails with mode F_P (segment starting with > and ending with ◇) only *after* $ac4_4$ ends. By looking at the figure, it is easy to see that such a delay is due to the failure with mode $F1$ of action $ac2_1$. The figure also shows actions $ac2_4$ and $ac3_2$, that are respectively a *put* and *load* related to another literal $at(o, loc4)$ (indicated by the ↗ after the actions), which fail as a consequence of the failure of $ac2_2$ (see Fig. 1). Also $ac3_4$ fails as a consequence of the failure of $ac3_2$.

[1]Note that, given a diagnosis δ, it is in general possible to find several alternative explanations, corresponding to different orders of events compatible with δ.

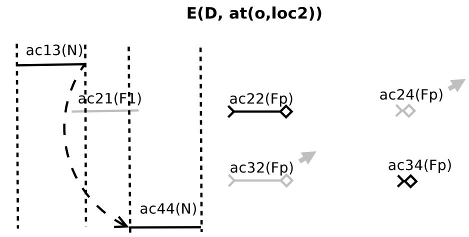

Figure 2: A Temporal Explanation of Diagnosis δ_{ex} = $\{ac2_1(F1)$, $ac2_2(F_P)$, $ac2_4(F_P)$, $ac3_2(F_P)$, $ac3_4(F_P)\}$.

	CBFS			
	time	#sol	time/sol	#expl
ag 2				
ac 8 (R2)	0.48	2.0	0.24	2.0
ag 4				
ac 10 (R2)	1.32	2.5	0.53	3.0
ac 20 (R2)	6.83	4.0	1.71	6.1
ac 20 (R4)	25.53	15.6	1.64	23.2

Table 1: avg time (sec), sols, time/sol, and expls in experiments.

4 IMPLEMENTATION AND TESTS

We have encoded the check of consistency of a hypothesis δ as an SMT problem for the solver Z3, and used a conflict-based best first search (CBFS) written in Java to find all the minimal rank diagnoses and explanations. For more details about CBFS see [9]. The tests have been run on a machine running Ubuntu 18.04.1 LTS, equipped with an i7 7700HQ CPU at 2.80 GHz, and 8 GB RAM. We have considered a *Logistic* domain where agents can *move*, *load*, and *put* objects, giving rise to several kinds of inter-agent interactions. We have experimented our approach by running a number of software simulated tests under different *configurations*, defined by varying the following dimensions: *#ag* (2 and 4 agents), *#ac* (8, 10, 20) actions per agent, *#rnk* (injected failures of ranks 2, 4). The observability rate (i.e., ratio between the number of actions with observable effects and the total number of actions) was 30%. In Table 1, we show results obtained with 4 different configurations of increasing complexity. The average total time for solving the PEF problems goes from 0.48s (2 agents x 8 actions, rank 2), up to 25.53s (4 agents x 20 actions, rank 4). However, it should be noted that the total time includes the computation of all the preferred diagnoses, as well as their temporal explanations. If we look at the average time taken for computing each preferred diagnosis (including its explanations), the increase is more limited, since, as the test cases become more challenging, the average number of preferred diagnoses increases (from 2.0 to 15.6), as well as the average number of associated explanations (from 2.0 to 23.2).

REFERENCES

[1] F. de Jonge, N. Roos, and C. Witteveen. 2009. Primary and secondary diagnosis of multi-agent plan execution. *Journal of Autonomous Agent and MAS* 18, 2 (2009), 267–294.

[2] Leonardo De Moura and Nikolaj Bjørner. 2008. Z3: An efficient SMT solver. In *International conference on Tools and Algorithms for the Construction and Analysis of Systems*. Springer, 337–340.

[3] M. Goldszmidt and J. Pearl. 1992. Rank-based systems: a simple approach to belief revision, belief update, and reasoning about evidence and actions. In *Proc. KR92*. 661–672.

[4] M. Kalech and G. A. Kaminka. 2007. On the design of coordination diagnosis algorithms for teams of situated agents. *Artificial Intelligence* 171, 8-9 (2007), 491–513.

[5] Roberto Micalizio and Pietro Torasso. 2014. Cooperative Monitoring to Diagnose Multiagent Plans. *Journal of Artificial Intelligence Research* 51 (2014), 1–70.

[6] Roberto Micalizio and Gianluca Torta. 2012. Diagnosing delays in multi-agent plans execution. In *Proceedings of the 20th European Conference on Artificial Intelligence*. IOS Press, 594–599.

[7] Roberto Micalizio and Gianluca Torta. 2016. Explaining interdependent action delays in multiagent plans execution. *Autonomous Agents and Multi-Agent Systems* 30, 4 (2016), 601–639.

[8] N. Roos and C. Witteveen. 2008. Diagnosis of Simple Temporal Networks. In *Proc. of ECAI'08*. 593–597.

[9] Gianluca Torta and Roberto Micalizio. 2018. SMT-Based Diagnosis of Multi-Agent Temporal Plans. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems, AAMAS 2018, Stockholm, Sweden, July 10-15, 2018*. 2097–2099.