SMT-Based Diagnosis of Multi-Agent Temporal Plans

Extended Abstract

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

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

ABSTRACT

The paper proposes a model and methodology for diagnosing action failures in the execution of Temporal Multi-Agent Plans (TMAPs). Contrary to previous proposals in the literature, we characterize actions with a finite set of possible execution modes, where each mode prescribes not only the logic post-conditions of the actions, but also an interval of possible durations.
Diagnoses are defined as assignments of modes to the actions that are consistent with the received observations and have the highest likelihood. We propose an algorithm that exploits a Satisfiability Modulo Theories (SMT) solver for the efficient computation of diagnoses. Preliminary experimental results are also presented.

KEYWORDS

Model-Based Diagnosis; Multi-Agent Plan; SMT

ACM Reference Format:
Gianluca Torta and Roberto Micalizio. 2018. SMT-Based Diagnosis of Multi-Agent Temporal Plans. In *Proc. of the 17th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2018), Stockholm, Sweden, July 10–15, 2018,* IFAAMAS, 3 pages.

1 INTRODUCTION

The diagnosis of the execution of a multi-agent plan (MAP) - i.e., a plan assigned to a team of (cooperating) agents - has been addressed in a number of works (see e.g., [1, 6, 7]), proposing different notions of plan diagnosis and different diagnostic methodologies. These works assume that the MAP is correct, but during its execution action failures may happen as a consequence of unexpected events such as faults in the functionalities of the agents, or unpredictable changes in the environment. All these works do not consider the temporal dimension; that is, action delays are never taken into account as possible causes and effects of plan execution anomalies. In many practical situations this appears to be a strong limitation since the MAP is often enriched with a schedule, and hence intermediate deadlines and the temporal constraints between actions have to be satisfied at execution time.

Some recent approaches have started to address the temporal dimension in the diagnosis of MAPs [8–10]. However, all these works do not tackle action failures that miss the achievement of some expected effects. In this abstract, we propose a novel formalism for capturing action failures *both* as missing effects (i.e., logic conditions), and as temporal delays.

Proc. of the 17th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2018), M. Dastani, G. Sukthankar, E. André, S. Koenig (eds.), July 10–15, 2018, Stockholm, Sweden.

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. To solve a diagnostic problem, we propose a methodology to infer the set of all the preferred diagnoses with *minimal rank* [4], i.e., with the highest (order-of-magnitude) likelihood. Specifically, given that we have to deal both with logic and temporal constraints to model faulty action modes, the computation of all the preferred diagnoses is made by exploiting a Satisfiability Modulo Theories (SMT) solver, that is able to handle both kinds of conditions. To the best of our knowledge, our proposal is the first one dealing with both temporal and logic aspects in the diagnosis of multi-agent plans. The most similar work we are aware of is [2], where, however, the authors concentrate only on conflicts among agents in the use of resources (e.g. road intersections).

2 FORMALIZATION

The major contribution of this abstract is the extension of the formal definition of Temporal Multi-Agent Plan (TMAP) in order to support the diagnosis task. In particular, we define a TMAP as the tuple $\langle T, A, R, C, M \rangle$, where:

- T is the team of cooperating agents $ag_1, ag_2, \ldots$.

- A is the set of action instances $ac_1, ac_2, \ldots$ included in the plan, each of which is assigned to a specific agent $agent(ac_i)$.

- R is a partial order relation over A establishing the precedence constraints in the execution of actions;

- C is a set of concurrency constraints of the form $\langle ac, ac' \rangle$, where $agent(ac) \neq agent(ac')$, and whose meaning is: the two actions are performed simultaneously as a *joint-action.*

- M is the set of all the possible *behavioral modes* that can be associated with the action instances in A. This is the novel element of our formalization that allows a knowledge engineer to define (anomalous) action behaviors according to the modality in which the action is performed. These behaviors encompass both delays in the completion of the action, as well as missing effects. Specifically, for each action ac a set of modes $M[ac]$ is defined. Due to space reasons, we omit a rigorous formalization of the modes in $M[ac]$. We just say that the designer can specify a common set *pre* of pre-conditions for action ac, and for each mode $m \in M[ac]$: (1) a time interval representing the possible range of duration of action ac while performed in modality m, and (2) a set *eff* of grounded literals resulting from the execution of ac. Therefore a modality is a (possibly faulty) action model, that is not used for the planning purpose, but for the diagnostic one, and hence it takes into account that actions may obtain different effects from the nominal, expected ones. More importantly, for each mode m the engineer has also to specify its *rank*: a non-negative integer value representing the

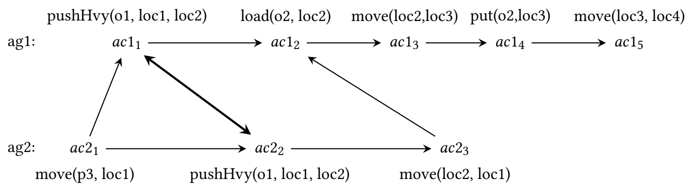

Figure 1: An example TMAP.

order-of-magnitude probability of the mode [4]: lower ranks correspond to higher probabilities. Rank zero is associated with all and only the nominal modalities.

Example 2.1. Let us consider a simple case with two agents: $T = \{ag1, ag2\}$ (see Figure 1). The set of actions is $A = \{ac1_1, \ldots, ac1_5, ac2_1, \ldots, ac2_3\}$, with order relations R as shown in the figure by the arrows. The double arrow between $ac1_1$ and $ac2_2$ denotes a joint-action: the two agent actions must be performed simultaneously to push a heavy object. The eight actions are instances of just four types of actions: *move*, *load*, *put*, and *pushHvy*. This last action can be performed only by two agents together. That is, in the TMAP in Figure 1, the two agents have to cooperate for moving a heavy object $o1$ from location $loc1$ to location $loc2$. Note the precedence relations between $ac2_1$ and $ac1_1$, and between $ac2_3$ and $ac1_2$. These relations impose that actions $ac1_1$ and $ac1_2$ can start only when action $ac2_1$ and $ac2_3$, respectively, have ended.

The TMAP model specifies also nominal and fault models. For instance, in nominal mode (N), a *move(ag,p1,p2)* requires the agent to be in place $p1$, causes the agent to arrive in place $p2$, and has an execution time in the interval $[1, 2]$. The rank is 0, meaning that the N mode is preferred in diagnostic explanations. In a faulty mode, say $F3$, the action has an execution time in the interval $[10, 25]$, a rank 3, and leaves the agent in $p1$.

3 PLAN EXECUTION FAILURE PROBLEM

We define a timed observation as a pair $\langle e, t\rangle$, where e is the observed event, and t is the time when e occurred.

Definition 3.1 (PEF problem). A Plan Execution Failure (PEF) problem is a pair $\langle P, Obs\rangle$ where P is a TMAP and Obs a set of timed observations.

We say that a mapping $H : A \rightarrow M[A]$ is a *hypothesis* about the modes of actions in P that assigns each action $ac \in A$ with a mode $m \in M[ac]$. Since action modes are associated with time intervals and logic pre-/post-conditions, a hypothesis H can be used to estimate a set of possible executions of P, that we call *temporal execution profiles*. We denote with $\mathcal{T}_P(H)$ the space of possible temporal execution profiles for the plan P consistent with H, and with $\mathcal{T}_P(Obs)$ the profile space consistent with timed observations Obs.

Definition 3.2 (PEF solution). Let $\langle P, Obs\rangle$ be a PEF problem, a solution to such a problem is a hypothesis H^{sol} such that:

(1) $\mathcal{T}_P(H^{sol}) \cap \mathcal{T}_P(Obs) \neq \emptyset$
(2) $rank(H^{sol})$ is minimal: no other hypothesis H' such that $rank(H') < rank(H^{sol})$ satisfies (1)

	CBFS		
	time	#sol	time/sol
ag 2			
ac 8	2.81	1.4	2
ag 3			
ac 10 (R1)	4.7	1.7	2.7
ac 10 (R2)	8.9	2.0	4.6
ag 4			
ac 10 (R2)	11.11	1.5	7.7
ac 20 (R2)	162	1.7	98.3
ac 20 (R4)	NA	NA	NA

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

Note that, as usual in a diagnostic setting, all the minimal solutions should be returned as an answer to a PEF problem.

Example 3.3. With reference to the TMAP in Figure 1, let us consider the set of observations Obs^1=$\{\langle holds(ag1, o2), 15\rangle\}$. In this case, three rank-one diagnoses are possible: it is sufficient to assume an $F1$ fault (which causes a delay in the interval $[3, 9]$) to either one of $ac2_1$, $ac2_2$, and $ac2_3$, to explain the delay of $ag1$ in leaving $o1$ at $loc1$. A rank-two diagnosis setting either $ac1_1$ or $ac1_2$ in mode $F2$ (delay $[10, 25]$) would also be consistent with Obs^1, but it is discarded as it is not minimal.

Now, let us consider observations Obs^2=$\{\langle holds(ag1, o2), 15\rangle, \langle at(ag_1, loc4), 15\rangle\}$; the observation about the position of agent ag_1 is sufficient to conclude that the only possible diagnosis is that action $ac1_4$ behaved with modality $F2$: the *put* of $o2$ by $ag1$ was unsuccessful in achieving the propositional effects. The latter case shows a diagnosis that not only explains a delay, but also an unexpected effect expressed as a propositional logic condition.

Given the encoding of a PEF problem in the input language of Z3, we compute the diagnoses exploiting the ability of Z3 to produce an *unsat core* every time it is invoked on an unsatisfiable instance. An unsat core is a set of assertions in the input to Z3 that cannot hold simultaneously and therefore require to withdraw at least one of them in order to get satisfiability. Given the set of unsat cores that is cumulatively produced during the search for the solutions, we can avoid to explore the parts of the search space that do not *hit* (i.e., withdraw at least an assignment from) all of them. Due to lack of space we omit details, but this technique is well known in diagnosis, also on approaches based on SMT [3, 5]. We will denote it as Conflict-based Best-First Search (CBFS).

4 EXPERIMENTAL EVALUATION

We have implemented the SMT-based approach to diagnosis described above as a Java program exploiting the Z3 solver. The tests were run on a virtual machine running Linux Ubuntu 14.04, equipped with an i7 M640 CPU at 2.80 GHz, and 4 GB RAM.

We have considered a *Logistic* domain which reflects the domain used in the examples. We have experimented our approach by running a number of software simulated tests under different *configurations*, defined by varying the number of agents *#ag* and number of actions *#ac*, and the ranks of injected failures.

In Table 1, we show results obtained with configurations including 2, 3, and 4 agents. The average times for finding a solution grow with the number of agents, number of actions per agent, and ranks of injected faults, as expected. We haven't reported results in the last row, since several cases took more than a 1 hour timeout on the test machine.

REFERENCES

[1] Femke de Jonge, Nico Roos, and Cees Witteveen. 2009. Primary and secondary diagnosis of multi-agent plan execution. *Journal of Autonomous Agent and Multi-Agent Systems* 18 (2009), 267–294.

[2] Orel Elimelech, Roni Stern, Meir Kalech, and Yedidya Bar-Zeev. 2017. Diagnosing resource usage failures in multi-agent systems. *Expert Systems with Applications* 77 (2017), 44–56.

[3] Juhan Ernits and Richard Dearden. 2011. Towards diagnosis modulo theories. In *22nd International Workshop on Principles of Diagnosis (DX-11)*. 249–256.

[4] Moisés Goldszmidt and Judea Pearl. 1992. Rank-based systems: a simple approach to belief revision, belief update, and reasoning about evidence and actions. In *Proceedings of the 3rd International Conference on Principles of Knowledge Representation and Reasoning (KR'92)*. 661–672.

[5] Alban Grastien. 2014. Diagnosis of hybrid systems with SMT: opportunities and challenges. In *Proceedings of the Twenty-first European Conference on Artificial Intelligence (ECAI'14)*. IOS Press, 405–410.

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

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

[8] Roberto Micalizio and Gianluca Torta. 2012. Diagnosing delays in multi-agent plans execution. In *Proceedings of the 20th European Conference on Artificial Intelligence (ECAI'12)*. IOS Press, 594–599.

[9] 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.

[10] Nico Roos and Cees Witteveen. 2008. Diagnosis of Simple Temporal Networks. In *Proceedings of the 18th European Conference on Artificial Intelligence (ECAI'08)*. 593–597.