Expert Systems With Applications 77 (2017) 44–56

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

Diagnosing resource usage failures in multi-agent systems

Orel Elimelech, Roni Stern*, Meir Kalech, Yedidya Bar-Zeev

Department of Software and Information Engineering, Ben Gurion University of the Negev, Be'er Sheva, Israel

ARTICLE INFO

Article history:
Received 19 May 2016
Revised 1 December 2016
Accepted 25 January 2017
Available online 2 February 2017

Keywords:
Model-based diagnosis
Multi-agent systems

ABSTRACT

In the not-so-far future, autonomous vehicles will be ubiquitous and, consequently, need to be coordinated to avoid traffic jams and car accidents. A failure in one or more autonomous vehicles may break this coordination, resulting in reduced efficiency (due to traffic load) or even bodily harm (due to accidents). The challenge we address in this paper is to identify the root cause of such failures. Identifying the faulty vehicles in such cases is crucial in order to know which vehicles to repair to avoid future failures as well as for determining accountability (e.g., for legal purposes). More generally, this paper discusses multi-agent systems (MAS) in which the agents use a shared pool of resources and they coordinate to avoid resource contention by agreeing on a *temporal resource allocation*. The problem we address, called the Temporal Multi-Agent Resource Allocation (TMARA) diagnosis problem (*TMARA-Diag*), is to find the root cause of failures in such MAS that are caused by malfunctioning agents that use resources not allocated to them. As in the autonomous vehicles example, such failures may cause the MAS to perform suboptimally or even fail, potentially causing a chain reaction of failures, and we aim to identify the root cause of such failures, i.e., which agents did not follow the planned resource allocation. We show how to formalize *TMARA-Diag* as a model-based diagnosis problem and how to compile it to a set of logical constraints that can be compiled to Boolean satisfiability (SAT) and solved efficiently with modern SAT solvers. Importantly, the proposed solution does not require the agents to share their actual plans, only the agreed upon temporal resource allocation and the resources used at the time of failure. Such solutions are key in the development and success of intelligent, large, and security-aware MAS.

1. Introduction

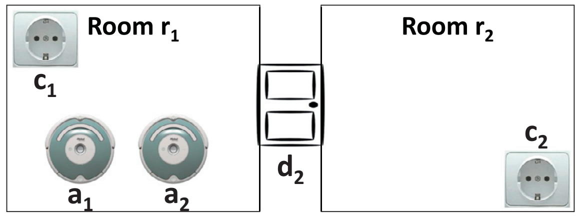
In many multi-agent systems (MAS) the agents require resources such as energy or space to perform actions and achieve their goals. The task of resource allocation in MAS is called Multi-Agent Resource Allocation (MARA). MARA is relevant in a wide range of applications including: industrial scheduling, network routing, airport traffic management, logistics and public transport (Chevaleyre et al., 2006). Some resources are consumed when they are used, e.g., fuel, while other resources can be reused multiple times by multiple agents. To coordinate the use of resources of the latter type, a *temporal* aspect is introduced into the resource allocation process, resulting in a *Temporal Multi-Agent Resource Allocation* (TMARA), which is a planned allocation of resources to agents over time. In a *proper* TMARA, resources are allocated to agents such that no resource is allocated to more than one agent.

Unfortunately, real agents may malfunction due to software or hardware failures and as a result use resources not according to the TMARA (De Jonge, Roos, & Witteveen, 2009). This faulty behavior may cause the entire MAS to fail when a resource allocated to an agent at some time is not available to it as planned, because it is used by the faulty agent.

As an example, consider the scenario depicted in Fig. 1. Two mobile robots, a_1 and a_2, are moving around between rooms (r_1 and r_2) and performing tasks. A doorway (d_1) connects the two rooms but it is only wide enough to allow a single robot to pass. Every hour, the robots need to recharge at one of the designated charging stations (c_1 and c_2), but, again, at most one robot can use a charging stations at a time. To avoid collisions in the doorways and long queues near the charging point, a TMARA, shown in the right side of Fig. 1, is generated to schedule when each robot is allowed to pass through the doorway or use a charging point. The TMARA indicates that a_1 is planned to pass through d_1 at time 1 and then charge in c_2 at time 2, while a_2 is planned to charge in c_1 at time 1 and then pass through d_1 at time 2. Now assume that a_1 fails and gets stuck in the doorway at time 1. In the next time step, a_2 will attempt to pass through d_1 and fail, because a_1 is currently occupying the door. In this paper we aim to *diagnose* such failed executions, identifying which agents were faulty and have caused this failure.

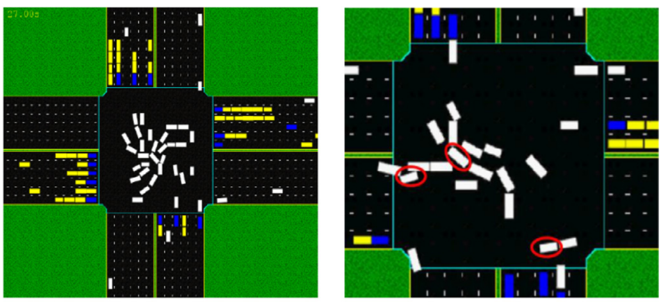
* Corresponding author.
E-mail addresses: orel.elim@gmail.com (O. Elimelech), roni.stern@gmail.com (R. Stern), kalech@bgu.ac.il (M. Kalech), ybz1984@gmail.com (Y. Bar-Zeev).

http://dx.doi.org/10.1016/j.eswa.2017.01.047
0957-4174/



TMARA			
Agents	1	2	Obs.
a_1	d_1	c_2	d_1
a_2	c_1	d_1	

Fig. 1. A simple scenario to demonstrate a TMARA (right) and a *TMARA-Diag* problem.



(a) Complex accident scenario. (b) A diagnosis.

Fig. 2. An example of an accident scenario in the AIM project Dresner and Stone (2008).

As MAS becomes more prevalent, there is a growing need for solving *TMARA-Diag* problems, because diagnosing the failure and identifying the faulty agents shorten the time required to fix the MAS and recover to a healthy state. Also, identifying the faulty agents is required to prevent future failures or to establish accountability for legal purposes.

The **first contribution** of this work is in formally defining this problem, which we call *TMARA-Diag*. Previous work on diagnosing MAS assumed that the agents' plans are given as input, imposing a partial order or temporal constraints over the agents' actions (De Jonge et al., 2009; Micalizio, 2009; Micalizio & Torasso, 2014; Micalizio & Torta, 2016; Roos & Witteveen, 2009). In other related work the explicit communication between the agents is monitored and used for diagnosis (Gutiérrez, 2013b; Gutierrez & García-Magariño, 2011). *TMARA-Diag* is fundamentally different from these prior work, as it is defined a TMARA, without having any additional knowledge about the agents' plans and without monitoring the communications between the agents. Thus, *TMARA-Diag* is especially suited for settings in which the agents are self-interested or privacy-aware, and only agree on sharing the TMARA. Therefore, our research is orthogonal and potentially complementary to prior work on diagnosis of MAS.

Solving a *TMARA-Diag* problem can be non-trivial, especially when the number of agents increases. For example, consider the intersection scenario depicted in Fig. 2. This example, taken from the Autonomous Intersection Management (AIM) project (Dresner & Stone, 2008) involved multiple driving agents aiming to pass the intersection as quickly as possible. There is an intersection manager agent that generates a TMARA, allocating to each driving agent a path through the intersection that is reserved for that agent for a particular time range. The left hand side of Fig. 2 shows an example of an accident that resulted by a single agent that did not follow the TMARA. Identifying which of the driving agents is responsible for this accident is exactly a *TMARA-Diag* problem, and solving it in this context is both important and difficult. The red circles in the right hand side of Fig. 2 shows three possible solutions – diagnoses – to this *TMARA-Diag* problem, which are three driver agents that may have caused the observed accident.

The **second contribution** of this work is in proposing a concrete and feasible approach to solve *TMARA-Diag* problems. Our approach is based on modeling *TMARA-Diag* as a model-based diagnosis (MBD) problem. MBD relies on a model of the diagnosed system, which is utilized to simulate the expected behavior of the system given the operational context (typically, the system inputs). The resulting simulated behavior (typically, the system outputs) are compared to the observed behavior of the system to detect discrepancies that indicate failures. The model is then used to pinpoint possible failing components within the system. In *TMARA-Diag*, the observed system behavior is the resources used by the agents when the MAS is observed after the failure. The system model in *TMARA-Diag* is the TMARA, as well as any additional knowledge available about how the agents behave. Modeling *TMARA-Diag* as an MBD enables solving *TMARA-Diag* problems by applying off-the-shelf MBD algorithms.

To solve *TMARA-Diag* problems effectively, two additional models are needed: a *conflict-model*, specifying how agents behave when they conflict, i.e., when they attempt to use the same resource, and a *fault-model*, specifying how faulty agents behave. Two possible conflict-models and two possible fault-models are presented and the resulting four configurations are analyzed. As a **third contribution**, we implemented an actual diagnosis algorithm for two of these configurations. The proposed diagnosis algorithm is based on compiling the *TMARA-Diag* problem to a Boolean satisfiability problem (SAT) using BEE (Ben-Gurion Equi-propagation Encoder) (Metodi, Codish, Lagoon, & Stuckey, 2011), an efficient constraint-to-CNF encoder. The experimental results demonstrate the feasibility of our diagnosis algorithm, which takes advantage of the power of modern SAT solvers.

2. Problem definition

In this section we formally define the *TMARA-Diag* problem.

Definition 1 (TMARA)**.** TMARA is a tuple $\langle A, R, T, bundle\rangle$ where A, R, and T, are the set of agents, resources and time steps, respectively, and $bundle$: $A \times T \rightarrow 2^R$ is a function that maps an agent $a \in A$ and time step $t \in T$ to the set of resources allocated to a for t.

In the example scenario in Fig. 4, A is the set of robots $\{a_1, a_2\}$, R is the set of doorways and charging points $\{d_1, c_1, c_2\}$, and $T = \{1, 2\}$. For every robot $a_i \in A$ and time step $t \in T$, the allocation function $bundle(i, t)$ returns the door or charging point, if any, a_i can use at time t.

A malfunctioning agent may use different resources than those assigned to it by the TMARA. Such a deviation from the planned TMARA may result in more than one agent simultaneously trying to use the same resource. In that case, these agents may not be able to continue their execution. This may cause a chain reaction, where other agents that do follow their TMARA are also unable to continue their execution, as the resources allocated to them in