Towards Model-Based Diagnosis of Coordination Failures

Meir Kalech and **Gal A. Kaminka***
The MAVERICK Group
Computer Science Department
Bar Ilan University, Israel
{kalechm,galk}@cs.biu.ac.il

Abstract

With increasing deployment of multi-agent and distributed systems, there is an increasing need for failure diagnosis systems. While successfully tackling key challenges in multi-agent settings, model-based diagnosis has left open the diagnosis of coordination failures, where failures often lie in the boundaries between agents, and thus the inputs to the model—with which the diagnoser simulates the system to detect discrepancies—are not known. However, it is possible to diagnose such failures using a model of the coordination between agents. This paper formalizes model-based coordination diagnosis, using two coordination primitives (concurrence and mutual exclusion). We define the consistency-based and abductive diagnosis problems within this formalization, and show that both are NP-Hard by mapping them to other known problems.

Introduction

Model-based diagnosis (*MBD*) (Reiter 1987; de Kleer & Williams 1987) relies on a model of the diagnosed system, which is utilized to simulate the behavior of the system given the operational context (typically, the system inputs). The resulting simulated behavior (typically, outputs) are compared to the actual behavior to detect discrepancies indicating failures. The model can then be used to pinpoint possible failing components within the system.

MBD is increasingly being applied in distributed and multi-agent systems (e.g., (Fröhlich *et al.* 1997; Roos, Teije, & Witteveen 2003; Lamperti & Zanella 2003)). While successfully addressing key challenges, MBD has been difficult to apply to diagnosing coordination failures (Micalizio, Torasso, & Torta 2004). This is because many such failures take place at the boundaries between the agent and their environment, including other agents. For instance, in a team, an agent may send a message that another agent, due to a broken radio, did not receive. As a result, the two agents come to disagree on an action to be taken. Lacking an omniscient diagnoser that knows of the sending of the message, the receiver has no way to detect and diagnose its fault, since the context—the message that can be fed into a model of the radio of both agents—is unobservable to the diagnoser.

*Gal Kaminka is also affiliated with CMU

Surprisingly, it is still often possible to detect and diagnose coordination failures, given the actions of agents, and the coordination constraints that should ideally hold between them. In the example above, knowing that the two agents should be in agreement as to their actions, and seeing that their actions are not in agreement, is sufficient to (1) show that a coordination failure has occurred; and (2) to propose several possible diagnoses for it (e.g., the first agent did not send a message, the second agent did not receive it, etc.).

Indeed, there are approaches to diagnosing such failures. However, they suffer from key limitations. Fault-based techniques (Horling, Benyo, & Lesser 2001; Pencolé, Cordier, & Rozé 2002; Lamperti & Zanella 2003) utilize pre-enumerated interaction fault models. When the faults are observed, they trigger possible predicted diagnoses. However, the interactions among system entities, in multi-agent system, are not known in advance since they depend on the specific conditions of the environment in runtime and the appropriate actions assigned by the agents (Micalizio, Torasso, & Torta 2004). (Kalech & Kaminka 2003) propose a technique in which the reasoning of the two agents, leading to their mis-coordinated actions, is re-traced, to determine the roots for their selection. However, this technique is specific to disagreements.

This work takes a first step towards addressing the open challenge of formalizing diagnosis of coordination (*inter-agent*) failures in terms of model-based techniques. We model the coordination between agents as a graph of concurrence and mutual exclusion constraints on agents' actions. The diagnosis process begins with an observation of the agents' actions and inferring, by comparing to the coordination model, the minimal number of agents that deviate from the expected coordination (i.e., a minimal set of *abnormal agents*).

The formalization allows definition of both consistency-based and abductive diagnosis problems, and points at several approaches to their solution. While the formalization does not commit to centralized or distributed diagnosis settings, the initial methods we provide are centralized. For consistency-based diagnosis, we show that computing the coordination diagnosis can be mapped to the minimal vertex cover problem. For abductive diagnosis, we take an approach based on satisfiability. Both of these problems are thus NP-Hard.

Related Work

(Pencolé, Cordier, & Rozé 2002; Lamperti & Zanella 2003) use a fault-model approach, where the distributed system is modeled as a discrete event system, and the faults are modeled in advance. The diagnoser infers unobservable fault events by computing possible paths in the discrete event system that match observable events. (Horling, Benyo, & Lesser 2001) and (Micalizio, Torasso, & Torta 2004) use causal models of failures and diagnoses to detect and respond to multi-agent and single-agent failures. A common theme in all of these is that they require pre-enumeration of faulty interactions among system entities. However, in multi-agent systems, these are not necessarily known in advance since they depend on the specific run-time conditions of the environment, and the actions taken by the agents.

(Fröhlich *et al.* 1997), and later (Roos, Teije, & Witteveen 2003) use a consistency-based approach to diagnose a spatially distributed systems. A set of n agents are responsible for diagnosing n sub-systems, correspondingly. Every agent makes a local diagnosis to its own sub-system and then all agents compute a global diagnosis. In order to build a global diagnosis set, each agent should consider the correctness of those inputs of its subsystem that are determined by other agents. But, Roos et al. and Fröhlich at al. assume that each diagnoser agent knows the context of its sub-system and so it may make the diagnosis. However in our multi-agent system the diagnoser does not have the context so it is impossible to make a diagnosis to every agent separately, unless we supply a model of the coordination between the agents.

(Kalech & Kaminka 2004) presented a consistency-based diagnosis procedure for behavior-based agents, which utilized a model of behaviors that the agents should be in agreement on (i.e., concurrence coordination). However, their approach was specific only to agreements.

Coordinated Multi-Agent Systems

We adopt a model-based diagnosis approach to diagnose the agents and the coordination failures. To do this, we formalize an agent, a multi-agent system, and the coordination between the agents.

The Agent Model

An agent is an entity that perceives its environment through sensors and takes actions upon its environment using actuators. Obviously, there are many different possible models that can be used to describe agents. Our focus is on the coordination of multiple agents through their actuators and their sensors, and thus we will use a simplified model, of completely reactive agents, composed only of sensor and actuator components. The connections between the sensors and actuators are described logically.

Definition 1. An *agent* is a pair $\langle CMP, CON \rangle$ of components CMP, and connections CON. CMP is a pair $\langle SEN, ACT \rangle$ where SEN is a set of boolean variables representing the sensors and the ACT is a set of boolean variables representing the actions. CON is a set of logical consequence statements, where the literals of SEN are on the left side of consequences, and the literals of ACT are on the right side.

At any time, the agent may sense through a number of sensors, but may only select one action. Thus multiple literals in SEN may be true, but at any time exactly one literal of ACT must be true. To enforce this, we apply a *completeness* formula (i.e. $ACT_1 \vee \ldots \vee ACT_{|ACT|}$) and a set of *mutual-exclusion* formulas $\forall i, j \neg(ACT_i \wedge ACT_j)$.

Example 1. Suppose we model a scout robot who looks for wounded. The robot has two sensor components, one is a radio sensor with two message values $\{seek, found\}$ and the other is a camera sensor which indicates if the wounded is found. The actions of the robot $\{SEEK, WAIT\}$ are selected based on the sensor readings: Once the robot receives a $seek$ message it selects the action $SEEK$. It will switch to the action $WAIT$ upon finding the wounded (via its camera), or upon receiving a message that it was found (by someone else). We represent this agent as follows:

$$
\begin{aligned}
SEN = \quad & \{SEN_{radio\_seek}, SEN_{radio\_found}, SEN_{camera\_found}\} \\
ACT = \quad & \{SEEK, WAIT\} \\
CON = \quad & \{SEN_{radio\_seek} \wedge \neg SEN_{camera\_found} \Rightarrow SEEK, \\
& SEN_{radio\_found} \vee SEN_{camera\_found} \Rightarrow WAIT\}
\end{aligned}
$$

In addition we should verify that only one action is selected by the agent, using the following *completeness* and *mutual-exclusion* axioms:

$$
\begin{gathered}
WAIT \vee SEEK \\
\neg(WAIT \wedge SEEK)
\end{gathered}
$$

A Model of Coordination

The multi-agent systems of interest to us are composed of several agents, which (by design) are to satisfy certain coordination constraints. We call this type of system a *team*, to distinguish it from general multi-agent systems in which it is possible that no coordination constraints exist.

Definition 2. A *team* T is a set of agents. $T = \{A_1...A_n\}$ where A_i is an agent. Given a team T, AS represents the set of the action literals of the agents. Formally, let ACT_i be the set of actions of agent A_i then $AS = \bigcup_{i=1}^{n} ACT_i$, where AS_{ij} represents the j'th boolean action variable of agent A_i. As a shorthand, we use AS_i to denote the boolean action literal of agent A_i whose value is true. We call AS_i the *active selection* of agent A_i.

The actions of agents in a team are coordinated. We utilize two coordination primitives—*concurrence* and *mutual exclusion*—to define the coordination constraints. Concurrence states that two specific actions must be taken jointly, at the same time. Mutual exclusion states the opposite, i.e., that two specific actions may not be taken at the same time.

Definition 3. A *concurrence coordination (CCRN)* constraint between two actions of different agents mandates that the two actions must be true concurrently. Logically, we represent this constraint in a DNF (disjunctive normal form). For two actions AS_{ix} and AS_{jy} (action x of agent A_i and action y of agent A_j) as follows:

$$CCRN(AS_{ix}, AS_{jy}) \Rightarrow (AS_{ix} \wedge AS_{jy}) \vee (\neg AS_{ix} \wedge \neg AS_{jy})$$

Definition 4. A *mutual exclusion coordination MUEX* constraint between two actions of different agents mandates that they cannot be true concurrently. Logically, for two actions AS_{ix} and AS_{jy},

$$
\begin{aligned}
MUEX(AS_{ix}, AS_{jy}) \quad \Rightarrow \quad & (AS_{ix} \wedge \neg AS_{jy}) \vee \\
& (\neg AS_{ix} \wedge AS_{jy}) \vee \\
& (\neg AS_{ix} \wedge \neg AS_{jy})
\end{aligned}
$$

Once we defined the coordination types, we can model the coordination between the agents formally with a set of coordination constraints, defining a graph:

Definition 5. A *coordination graph* for a team T is an undirected graph $CG = \{V, E\}$, where the vertices set V represents the boolean variables of the actions of the agents, and the set of edges E is the set of coordination constraints between the actions. We use CG_m to refer to the m'th constraint within E. $CG(AS_{ix}, AS_{jy})$ denotes the constraint relating AS_{ix} and AS_{jy}. CG_m is considered true if the constraint holds and false otherwise.

Example 2. Figure 1 presents a coordination graph. The concurrence constraints are represented by solid lines, and the mutual exclusion constraints are represented by dashed lines. Assume a team of three agents $\{A_1, A_2, A_3\}$. A_1 and A_2 are scout robots as described in Example 1, and A_3 is a paramedic robot who has one radio sensor with one message value *{found_message}*, and three actions *{JOIN, TREAT, CHARGE}*. Agents A_1 and A_2 have the same role in the team so they have *concurrence coordination* constraints between their actions. At the beginning A_1 and A_2 receive a $seek$ message so they select the action $SEEK$ while A_3 may select any action except $TREAT$, meaning it can not treat a wounded, while the other robots seek. We can see the *mutual exclusion coordination* constraints between these behaviors. Once A_1 or A_2 find the wounded, they send a $found\_message$ to the other agents in the team, then A_1 and A_2 transport to the $WAIT$ action, while A_3 transports to $JOIN$ action. Again we can see the *concurrence coordination* constrains between these behaviors. In addition, when agent A_3 is being charged ($CHARGE$ behavior), there are no constraints between the agents. The corresponding CG is formally defined as follows:

$$
\begin{aligned}
V = \;& \{AS1_{WAIT}, AS2_{WAIT}, AS1_{SEEK}, AS2_{SEEK},\\
& AS3_{TREAT}, AS3_{JOIN}, AS3_{CHARGE}\}\\
E = \;& \{CCRN(AS1_{WAIT}, AS2_{WAIT}),\\
& CCRN(AS1_{SEEK}, AS2_{SEEK}),\\
& MUEX(AS2_{SEEK}, AS3_{TREAT}),\\
& CCRN(AS2_{WAIT}, AS3_{JOIN}),\\
& MUEX(AS1_{SEEK}, AS3_{TREAT}),\\
& CCRN(AS1_{WAIT}, AS3_{JOIN})\}
\end{aligned}
$$

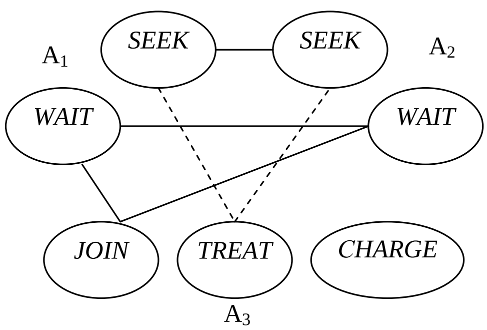

Figure 1: **The coordination graph for team $\{A_1, A_2, A_3\}$.**

Given a coordination graph CG and a team T, we can define a multi-agent system description as a set of implications from the normality of the agents to the satisfaction of the coordination constraints. This is the final piece in formalizing a normally-functioning multi-agent system.

Definition 6. A *multi agent system description (MASD)* is a set of implications from the normality of agents in a team T to CG. The meaning of the predicate $AB(\cdot)$ is that the corresponding agent is considered abnormal (failing).

$$
\begin{aligned}
MASD = \;& \{\neg AB(A_i) \wedge \neg AB(A_j) \Rightarrow CG(AS_{ix}, AS_{jy})\\
& |CG(AS_{ix}, AS_{jy}) \in CG \wedge A_i, A_j \in T\}
\end{aligned}
$$

Diagnosis of Coordination Faults

A fault in the coordination of a multi-agent system may be the result of a fault in one of the sensors or other agent components [1] Given an $MASD$ and a set of normality assumptions, it is possible to infer that a fault exists (and to generate hypotheses as to its identity), by checking whether the observed actions of the agents satisfy the $MASD$.

Let us formalize the coordination diagnosis in terms of model based diagnosis:

Definition 7 *Coordination Diagnosis Problem*. Given $\{T, MASD, AS\}$ where T is a team of agents $\{A_1...A_n\}$, $MASD$ is a multi agent system description defined over T (Def. 6), and AS is the set of the actions of the agents (Def. 2), then the *coordination diagnosis problem (CDP)* arises when

$$MASD \cup \{\neg AB(A_i) | A_i \in T\} \cup AS \vdash \perp$$

We use the following example to illustrate.

Example 3. Suppose we are given the following $MASD$, T, and AS (only the true literals in AS are shown):

$$
\begin{aligned}
T = \;& \{A_1, A_2, A_3, A_4, A_5, A_6\}\\
MASD = \;& \{\neg AB(A_1) \wedge \neg AB(A_4) \Rightarrow MUEX(AS_{11}, AS_{41}),\\
& \neg AB(A_1) \wedge \neg AB(A_2) \Rightarrow CCRN(AS_{12}, AS_{21}),\\
& \neg AB(A_1) \wedge \neg AB(A_6) \Rightarrow CCRN(AS_{12}, AS_{61}),\\
& \neg AB(A_2) \wedge \neg AB(A_3) \Rightarrow CCRN(AS_{22}, AS_{31}),\\
& \neg AB(A_2) \wedge \neg AB(A_5) \Rightarrow CCRN(AS_{22}, AS_{51}),\\
& \neg AB(A_2) \wedge \neg AB(A_6) \Rightarrow CCRN(AS_{21}, AS_{61}),\\
& \neg AB(A_3) \wedge \neg AB(A_4) \Rightarrow MUEX(AS_{32}, AS_{42}),\\
& \neg AB(A_3) \wedge \neg AB(A_5) \Rightarrow CCRN(AS_{31}, AS_{51})\}\\
AS = \;& \{AS_{11}, AS_{21}, AS_{31}, AS_{41}, AS_{51}, AS_{61}\}
\end{aligned}
$$

Figure 2 shows the coordination graph for this CDP. Assuming all the agents are not abnormal, the actions of the agents are not consistent with the coordination graph. For instance, the actions $AS_{11} = true$ and $AS_{41} = true$ causes an inconsistency in CG_1, as it produces a false value of $MUEX(AS_{11}, AS_{41})$, though, $MUEX(AS_{11}, AS_{41})$ should be true, given the normality

[1] It may also be the result of a fault in the environment, e.g., when a message is lost in transit. This is treated as a fault in the receiver.

assumptions $\neg AB(A_1), \neg AB(A_4)$. On the other hand, if the actions $AS_{12}, AS_{21}, AS_{32}, AS_{41}, AS_{52}, AS_{61}$ were true (implying that the other actions were false), they would have been consistent with the coordination graph.

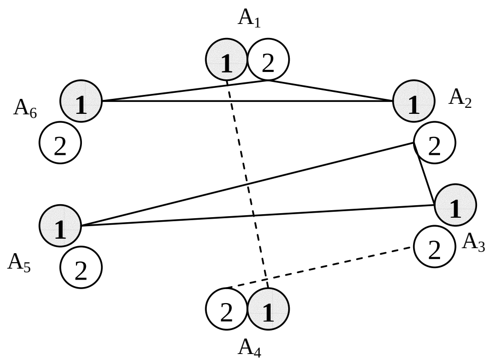

Figure 2: **The coordination graph and active selection (gray circles) of the team** $T = \{A_1, A_2, A_3, A_4, A_5, A_6\}$

Given a CDP, the goal of the coordination diagnosis process is to determine a minimal set of abnormal agents whose selection and subsequent setting of the $AB(\cdot)$ clause would eliminate the inconsistency (consistency-based diagnosis, Section), or explain it (abductive diagnosis, Section). A coordination diagnosis (a set of abnormal agents) is minimal, iff no proper subset of it is a coordination diagnosis.

Once the set of such abnormal agents is found, the diagnoser infers the abnormal components (in our case, sensors) within the abnormal agents. This is done using straightforward back-chaining through the set CON (Def. 1) of logical consequence statements connecting sensors to actions (e.g., as in (Kalech & Kaminka 2003)).

Consistency-Based Coordination Diagnosis

We begin by defining consistency-based coordination diagnosis.

Definition 8. A *consistency-based global coordination diagnosis (CGCD)* is a minimal set $\Delta \subseteq T$ such that:

$$MASD \bigcup \{AB(A_i)|A_i \in \Delta\} \bigcup \{\neg AB(A_i)|A_i \in T - \Delta\} \bigcup AS \nvdash \bot$$

The first step in this process to determine which agents are in conflict:

Definition 9. Two agents a and b are called *conflict pair* $\langle a, b\rangle$, if there exist a constraint CG_i that relates a and b and whose value is false.

$$\forall a, b \in T, \exists i, j, k \; s.t. \neg CG_i(AS_{aj}, AS_{bk}) \Rightarrow \langle a, b\rangle$$

Definition 10. A *local conflict set* is a set of the all conflict pairs in the system, and is denoted by LC.

Example 4. LC in the graph of example 3 is: $LC = \{\langle A_1, A_4\rangle \, \langle A_1, A_2\rangle, \langle A_1, A_6\rangle, \langle A_2, A_3\rangle, \langle A_2, A_5\rangle\}$

The local conflict set forms the basis for the $CGCD$, because for each conflict pair, at least one of the agents is abnormal. However, the $CGCD$ is not a simple combination of all agents in the LC pairs, as arbitrary selection of agents may lead to diagnosis sets that are themselves inconsistent. For instance, treating each pair in the computed LC in Example 4 by itself, produces the following subset of possible diagnoses:

$$\langle A_1, A_2\rangle \Rightarrow \{AB(A_1), \neg AB(A_2)\}$$
$$\langle A_1, A_2\rangle \Rightarrow \{\neg AB(A_1), AB(A_2)\}$$
$$\langle A_1, A_4\rangle \Rightarrow \{AB(A_1), \neg AB(A_4)\}$$
$$\langle A_1, A_4\rangle \Rightarrow \{\neg AB(A_1), AB(A_4)\}$$

It is easy to see that combining these diagnoses may produce inconsistency (for instance, combining the first and last implications would produce the set $\{AB(A_1), \neg AB(A_2), \neg AB(A_1), AB(A_4)\}$).

Therefore, we cannot diagnose every conflict pair by itself and then combine the results. Rather, we should compute the diagnoses sets Δ considering the dependencies between the conflict pairs. To do this, we should look for the abnormal agent(s) in every conflict pair.

We achieve this goal by generating a hitting-set of agents, selecting at least one agent as abnormal from every conflict pair, such that the resulting agents cover between them all conflict pairs. We want to maintain a minimal number of such agents. This is somewhat similar to Reiter's HS-Tree (1987), or de Kleer and Williams' technique (1987). It is also related to minimal model techniques used in non-monotonic reasoning (Olivetti 1992; Niemelä 1996). We plan to explore these connections in the future.

We achieve this goal by transforming the conflict set into a graph, and finding the vertex cover for this graph. Let us define a conflict graph $G = \{V', E'\}$ where E' is a set of the conflict pairs and V' is a set of the agents involved in the conflict set. In order to compute the diagnosis we run an algorithm to find a minimal vertex cover—a set of vertices that involve all edges. A vertex cover set is guaranteed to be a diagnosis since all the edges, namely the conflict pairs, are covered by this set, namely by a set of abnormal agents. We are looking for all the possible minimal vertex cover sets, since the diagnosis contains all the possibilities of abnormal agents. Minimal vertex covers guarantee minimal diagnosis, since a vertex cover is minimal only if no proper subset of it is a vertex cover.

Determining a minimal vertex cover is known to be NP-Complete, however the problem of determining the set of minimal vertex covers is NP-Hard (Skiena 1990). A simple $O(2^{|V|})$ exact algorithm for its solution is to find all the possible vertex covers in size one, then continue to find the possible vertex cover in size two, under the condition that it is not a superset of a previous vertex cover, and so on up to the max size of the graph. The complexity of computing the $CGCD$ is thus the same as in single-agent diagnosis methods, e.g., (de Kleer & Williams 1987).

Example 5. Figure 3 presents the graph of the conflict pairs that were computed in example 4. The vertex cover set of size one is empty, for size two it is $VC_1 = \{A_1, A_2\}$, and there are two sets of size three: $VC_2 = \{A_1, A_3, A_5\}$ and $VC_3 = \{A_2, A_4, A_6\}$ (there are more vertex cover sets which are superset of VC_1), it is unnecessary to continue to check the vertex cover in size four and more since every such vertex cover will be a superset of the formers. By building the vertex cover sets we obtain the global coordination diagnosis, $\Delta_1 = \{A_1, A_2\}$, $\Delta_2 = \{A_1, A_3, A_5\}$, $\Delta_3 = \{A_2, A_4, A_6\}\}$.

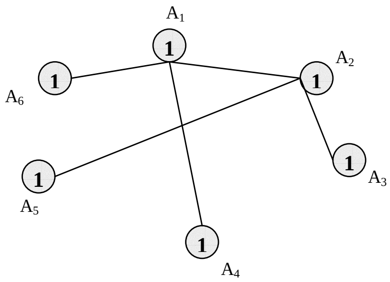

Figure 3: **A graph of the conflict pairs in example 4**

A disadvantage of the consistency-based approach is that it may produce diagnoses that are unsound, in the sense that while they eliminate the inconsistency, they do not explain it. Intuitively, such diagnoses correspond to eliminating the abnormal agents from consideration, rather than suggesting that they change their actions. For such diagnoses, there may be no actions that the abnormal agents could take that would be consistent with the $MASD$.

For instance, in Example 5 the diagnosis set $\{A_1, A_2\}$ represents a minimal set of abnormal agents, but changing their actions $(A_{11} = false, A_{12} = true, A_{21} = false, A_{22} = true)$ will leave the system inconsistent, with $CCRN(AS_{12}, AS_{21}) = false$. On the other hand, changing the actions of the agents in the other diagnoses $(\{A_1, A_3, A_5\}, \{A_2, A_4, A_6\})$ will eliminate the inconsistency.

Abductive Coordination Diagnosis

The implication is that stronger conditions on the solution sets may be needed. Such conditions correspond to abductive diagnosis, in which changing the actions of the abnormal agents entails the coordination graph:

Definition 11. An *abductive global coordination diagnosis (AGCD)* is a minimal set $\Delta \subseteq T$ such that:

$$MASD \bigcup \{AB(A_i)|A_i \in \Delta\} \bigcup \{\neg AB(A_i)|A_i \in T - \Delta\} \bigcup AS \nvdash \perp$$

and,

$$\{AB(A_i)|A_i \in \Delta\} \bigcup \{\neg AB(A_i)|A_i \in T - \Delta\} \bigcup AS \Rightarrow CG$$

where, we make the active selection of agent A_i (Def. 2), AS_i, false, and force A_i to choose a different action,

$$AB(A_i) \Rightarrow \neg AS_i \wedge (AS_{i1} \vee \ldots \vee AS_{i|ACT|})$$

The first condition in Def. 11 is exactly as in Definition 8 (i.e., $CGCD$) to satisfy the consistency requirement. The second condition requires that for any abnormal agents found, it will be possible to change their active selection, in order to entail the coordination graph and thus satisfy the coordination constraints. Note that the entailment here is of the coordination graph, not the full $MASD$.

The unsound diagnosis set $\{A_1, A_2\}$, given by the consistency-based approach, will not pass this second condition, since the alternative actions of agent A_1 and of agent A_2 do not entail the coordination graph.

In order to satisfy Definition 11, the diagnosis process needs to go beyond pinpointing suspect agents, to verifying that by changing their actions, coordination will be restored. Thus in contrast with consistency-based approach, we do not utilize conflict pairs to compute the diagnoses, but instead examine all action literals assignments that entail the coordination graph, i.e., all actions which will satisfy the coordination constraints. Then the process compares the existing truth values to those that will satisfy the coordination, and computes a *minimal* set of changes.

Example 6. Let us compute the $AGCD$ of the Example 3. Table 1 presents the satisfying truth assignments for the actions of agents $A_1 \ldots A_6$. There are only two such assignments. In order to find the minimal *AGCD*, we should compare the actions of the agents with these assignments and point out the agents that deviate. Consider the actions in Example 3 (where $AS_{11}, AS_{21}, AS_{31}, AS_{41}, AS_{51}, AS_{61}$ are true, and the other action literals are false). Then, in the first row $AS_{11} = false$, but we have $AS_{11} = true$. We thus mark action AS_{11} as faulty. The second value in the table is $AS_{12} = true$, but we have $AS_{12} = false$, so we again mark this as faulty, and so on for each one of the actions. For the first entry in the truth table we got the following faulty actions: $AS_{11}, AS_{12}, AS_{31}, AS_{32}, AS_{51}, AS_{52}$. From this list, we can determine the abnormal agents by finding the agents whose actions are faulty. We thus conclude that a minimal *AGCD* is $\Delta_1 = \{A_1, A_3, A_5\}$ for this row. From the second row, we similarly find $\Delta_2 = \{A_2, A_4, A_6\}$. Setting these agents to abnormal, and thus forcing them to select different actions, would satisfy the coordination constraints.

#	A_1		A_2		A_3		A_4		A_5		A_6	
	1	**2**	**1**	**2**	**1**	**2**	**1**	**2**	**1**	**2**	**1**	**2**
1	0	1	1	0	0	1	1	0	0	1	1	0
2	1	0	0	1	1	0	0	1	1	0	0	1

Table 1: **Coordination-satisfying actions in Example 6.**

Obviously, we should consider only the minimal *AGCD*. We fulfill this requirement by comparing every new hypothesized coordination diagnosis to the former coordination diagnoses, and checking whether it is a subset, a superset, or different than the former diagnoses.

Thus the $AGCD$ problem is essentially that of finding all sets of truth assignments that will satisfy a target proposition, an NP-Hard problem. A detailed discussion of satisfiability, and the rich literature offering efficient exact and approximate solution methods is well beyond the scope of this paper. However, we point at two diagnosis-specific mechanisms that can potentially be used to alleviate computational load in our case:

1. Ordered binary decision diagram (OBDD) (Bryant 1992) can be used to efficiently reason about diagnosis-satisfying assignments (Torasso & Torta 2003). By restricting the representation, boolean manipulation becomes much simpler computationally. We can compactly represent the coordination graph using OBDDs (an offline construction process), and then truth assignments can be computed in linear time in many cases.

2. Assumption-based truth maintenance systems (ATMS) (de Kleer 1986) can be used to build the satisfying assignments incrementally. We exploit the fact that it is unnecessary to check all the assignments since the legal assignments depend each on the other. For instance, assume a concurrence coordination between a and b and between b and c:

$$((a \wedge b) \vee (\neg a \wedge \neg b)) \bigwedge ((b \wedge c) \vee (\neg b \wedge \neg c))$$

   Instead of computing the full truth table of a, b and c, (2^3), we can use an ATMS, which given these justifications will provide only two assignments: $(a = true, b = true, c = true)$ or $(a = false, b = false, c = false)$.

There is also an obvious connection between this problem and constraint satisfaction and optimization problems, though the $AGCD$ process looks for *all* minimal solutions, rather than any minimal solution. We plan to explore this in the future.

Summary and Future Work

We presented a novel formalization for diagnosing coordination failures in multi agent systems without knowing the inputs of the agents, instead relying on a model of the coordination between the agents. We model such coordination using two coordination primitives (concurrence and mutual exclusion). In the diagnosis process the diagnoser observes, the actions of the agents, then it finds the candidate abnormal agents by the coordination model, and finally continues to compute the abnormal sensors by back-chaining (shown in (Kalech & Kaminka 2003)).

We defined both a consistency-based and abductive diagnosis versions of coordination diagnosis, and proposed initial algorithms for both. The consistency-based approach finds the local conflicts between pairs of agents, then continues to compute the diagnosis by combining the conflicts using a minimal vertex cover algorithm. We showed that this approach is unsound, in that it may produce diagnoses that are impossible, in that they cannot be corrected. The second approach maps the abductive coordination diagnosis problem to that of satisfiability, finding a minimal set of truth-value changes that satisfy a given proposition. Here, our initial approach pre-computes all the possible coordination-satisfying action assignments, and then uses these during online diagnosis by comparing the actions of the agents to each one of the instances of the satisfying action assignments.

Our goal in this paper was to take a first step towards the use of MBD techniques in multi-agent systems. Much is left for future research. First, the algorithms we proposed are related to key techniques in diagnosis, CSP, and non-monotonic reasoning. We plan to explore the connections, to bring to bear on this diagnosis problem. Second, the presented model is sensors-actions based since we would like to focus on the coordination diagnosis issue. However, as it stands, it can already be used to model simple situated agents, in domains such as RoboCup and others (Kalech & Kaminka 2003). We hope to explore richer models of complex multi-component agent models, and richer coordination primitives in the future. In addition, while this paper has adopted the perspective of a centralized single diagnoser, we plan to tackle distributed algorithms next.

Acknowledgments

This research was supported in part by BSF grant 2002401. We gratefully acknowledge a generous gift from IBM. As always, thanks to K. Ushi and K. Ravit.

References

Bryant, R. E. 1992. Symbolic Boolean manipulation with ordered binary-decision diagrams. *ACM Computing Surveys* 24(3):293–318.

de Kleer, J., and Williams, B. C. 1987. Diagnosing multiple faults. *Artificial Intelligence* 32(1):97–130.

de Kleer, J. 1986. An assumption-based truth maintenance system. *Artificial Intelligence* 28:127–162.

Fröhlich, P.; de Almeida Mora, I.; Nejdl, W.; and Schröder, M. 1997. Diagnostic agents for distributed systems. In *ModelAge Workshop*, 173–186.

Horling, B.; Benyo, B.; and Lesser, V. 2001. Using Self-Diagnosis to Adapt Organizational Structures. *Proceedings of the 5th International Conference on Autonomous Agents* 529–536.

Kalech, M., and Kaminka, G. A. 2003. On the design of social diagnosis algorithms for multi-agent teams. *in International Joint Conference on Artificial Intelligence (IJCAI-03)* 370–375.

Kalech, M., and Kaminka, G. A. 2004. Diagnosing a team of agents: Scaling-up. *In Proceedings of the International Workshop on Principles of Diagnosis (DX 2004)* 129–134.

Lamperti, G., and Zanella, M. 2003. *Diagnosis of Active Systems*. Kluwer Academic Publishers.

Micalizio, R.; Torasso, P.; and Torta, G. 2004. On-line monitoring and diagnosis of multi-agent systems: a model based approach. *in Proceeding of European Conference on Artificial Intelligence (ECAI 2004)* 16:848–852.

Niemelä, I. 1996. A tableau calculus for minimal model reasoning. In *Proceedings of the fifth workshop on theorem proving with analytic tableaux and related methods.*

Olivetti, N. 1992. A tableaux and sequent calculus for minimal model entailment. *Journal of automated reasoning* 9:99–139.

Pencolé, Y.; Cordier, M.; and Rozé, L. 2002. Incremental decentralized diagnosis approach for the supervision of a telecommunication network. *IEEE Conference on Decision and Control (CDC'02)* 435–440.

Reiter, R. 1987. A theory of diagnosis from first principles. *Artificial Intelligence* 32(1):57–96.

Roos, N.; Teije, A. t.; and Witteveen, C. 2003. A protocol for multi-agent diagnosis with spatially distributed knowledge. *in Proceedings of Autonomous Agents and Multi Agent Systems (AAMAS-03)* 655–661.

Skiena, S. 1990. *Implementing Discrete Mathematics: Combinatorics and Graph Theory with Mathematica*. Addison-Wesley.

Torasso, P., and Torta, G. 2003. Computing minimum-cardinality diagnoses using OBDDs. *Advances in Artificial Intelligence (lecture notes in artificial intelligence)* 2281:224–238.