Gero Iwan

Aachen University of Technology
Department of Computer Science V
52056 Aachen, Germany
iwan@informatik.rwth-aachen.de

August, 1999

Agents who carry out a course of actions inevitably run into the problem that things do not work out as planned. For example, a robot delivering a book may end up losing the book along the way or delivering it to the wrong room. Finding out what went wrong and recovering from it is a difficult and largely unsolved problem.

In contrast to traditional work on diagnosis where the focus is on a static analysis of "what is wrong", diagnosis in settings like mobile robots acting in a changing environment focusses on "what happened" which we refer to as *history-of-events diagnosis*.

Given a description of system behaviour and an (assumed) history of occurred events the diagnostic task arises from a contradicting observation. In [McI98] so-called *explanatory diagnoses* are studied, which are continuations of the history explaining the observation. It is shown that this kind of diagnosing is analogous to planning.

In our approach to diagnosis we allow adding of events not only at the end but at any point of the history. In addition to that we exploit another source of explanation by taking into account the possibility that some history events might not have happened as assumed (or might not have occurred at all). Obviously, in environments with uncertain knowledge about occurrence and outcome of events this kind of reasoning is very important, as is the former one. So both have to be combined.

As an example we look at an autonomous robot, whose task it is to bring book B into room R. Suppose the robot and the book are in the same room already. The robot decides (plans) to carry out the sequence of actions

$$\vec{\alpha} = [\mathit{pick\_up}(\mathsf{B}), \mathit{start\_for}(\mathsf{R}), \mathit{arrive\_at}(\mathsf{R}), \mathit{put\_down}(\mathsf{B})]$$

and initiates its execution. In the situation attained after the (assumed) execution of the four actions $\vec{\alpha}$ is the (assumed) history, and it is derivable, that B ought to be in R. Now the robot receives the message (e. g., by the disappointed would-be recipient), that B is not in R. This contradicts the assumed history $\vec{\alpha}$. But what happened actually? Some explanations are:

(1) The robot lost B on its way to R.

(2) The robot lost its way and entered room R′ instead of R.

(3) The robot failed to grip B during the *pick_up*-action.
So a "failure variation" of *pick_up*, say *pick_up′*, happened instead of the "real" *pick_up*-action.

(4) Somebody took away B after the robot had put it down in R.

To the given four explanations correspond four diagnoses which are modified histories explaining the fact that B is not in R:

$$\vec{\gamma}_{(1)} = [\mathit{pick\_up}(\mathsf{B}), \mathit{start\_for}(\mathsf{R}), \mathit{robot\_loses}(\mathsf{B}), \mathit{arrive\_at}(\mathsf{R}), \mathit{put\_down}'(\mathsf{B})]$$

$$\vec{\gamma}_{(2)} = [\mathit{pick\_up}(\mathsf{B}), \mathit{start\_for}(\mathsf{R}), \mathit{arrive\_at}(\mathsf{R}'), \mathit{put\_down}(\mathsf{B})]$$

$$\vec{\gamma}_{(3)} = [\mathit{pick\_up}'(\mathsf{B}), \mathit{start\_for}(\mathsf{R}), \mathit{arrive\_at}(\mathsf{R}), \mathit{put\_down}'(\mathsf{B})]$$

$$\vec{\gamma}_{(4)} = [\mathit{pick\_up}(\mathsf{B}), \mathit{start\_for}(\mathsf{R}), \mathit{arrive\_at}(\mathsf{R}), \mathit{put\_down}(\mathsf{B}), \mathit{somebody\_takes}(\mathsf{B})]$$

In cases (1) and (3) the "real" *put_down*-action could not have taken place since it is necessary to have an object in order to put it down. (Note, that only $\vec{\gamma}_{(4)}$ is a continuation of $\vec{\alpha}$.)

Of course, there are many other explanations resp. diagnoses, e. g.

$$\vec{\gamma}_{(5)} = [\mathit{pick\_up}(\mathsf{B}), \mathit{start\_for}(\mathsf{R}), \mathit{arrive\_at}(\mathsf{R}'), \mathit{put\_down}(\mathsf{B}), \mathit{somebody\_takes}(\mathsf{B})]$$

$$\vec{\gamma}_{(6)} = [\mathit{pick\_up}(\mathsf{B}), \mathit{start\_for}(\mathsf{R}), \mathit{arrive\_at}(\mathsf{R}), \mathit{put\_down}(\mathsf{B}), \mathit{pick\_up}(\mathsf{B}), \mathit{start\_for}(\mathsf{R}'), \mathit{arrive\_at}(\mathsf{R}'), \mathit{put\_down}(\mathsf{B})]$$

However, $\vec{\gamma}_{(5)}$ is a continuation of $\vec{\gamma}_{(2)}$ (and hence *somebody_takes*(B) is superfluous). $\vec{\gamma}_{(6)}$ should not be considered a desirable diagnosis since the robot would have known if it had brought B into R′ after bringing it into R (and therefore it had assumed $\vec{\gamma}_{(6)}$ to be the history instead of $\vec{\alpha}$).

From this simple scenario we can already infer the following requirements: a diagnosis should

- form a possible history (according to the given description of system behaviour)
- explain the observation
- take into consideration the so far assumed history, i. e., include (in the corresponding order) all history events/actions or *variations* of them (because there may be uncertainty about the actual effects of events/actions, but not about their initiation)[1]
- use as additional events/actions only suitable *"explanatory events/actions"*, i. e., such events/actions which are not under the agent's control but may have occurred and can help to explain the observation
- be parsimonious (e. g., avoid events/actions, which do not contribute to the explanation or are otherwise superfluous)[2]

The framework for our theoretical investigations on the subject is the situation-as-histories variant of the *situation calculus* [LPR98], enriched with an unary predicate denoting **explanatory actions** and a binary predicate denoting the **variation** relation between actions. An **observation** $\phi(s)$ is simply a situation calculus formula with the situation variable s as only free variable.

Let $\vec{\alpha} = [\alpha_1, \ldots, \alpha_n]$ and $\vec{\gamma} = [\gamma_1, \ldots, \gamma_m]$ be sequences of actions. $\vec{\gamma}$ is an **explanatory variation** of $\vec{\alpha}$ iff there exists $1 \leq i_1 < \ldots < i_n \leq m$ such that γ_{i_j} is α_j or a variation of α_j and all the other γ_i are explanatory actions. $\vec{\gamma}$ is a **history-of-events diagnosis** for $\vec{\alpha}$ and $\phi(s)$ iff $\vec{\gamma}$ is an executable explanatory variation of $\vec{\alpha}$ and $\phi(\sigma)$ holds where σ is the situation attained after the execution of $\vec{\gamma}$.

This definition can be formulated as a situation calculus formula which captures all of the above-mentioned requirements except the last. A subsequence test yields a simple preference criterion. However, more elaborate criteria are necessary, possibly based on preferences and probabilities of action variations and explanatory actions (e. g., for a robot with a reliable gripper but inaccurate navigation, losing the way is more likely than losing an object on the way).

The topics currently under investigation include

- *diagnosis preference criteria*
- *detecting diagnosis representations*
  E. g., for each room different from R there is a diagnosis similar to $\vec{\gamma}_{(2)}$. They all could be represented by the *diagnosis pattern*
  $$\vec{\mu} = [\mathit{pick\_up}(\mathsf{B}), \mathit{start\_for}(\mathsf{R}), \mathit{arrive\_at}(\,r\,), \mathit{put\_down}(\mathsf{B})]$$
  together with the constraint $r \neq \mathsf{R}$.
- *incorporating intermediate observations*
  E. g., if the robot had checked that it had B before heading towards R, $\vec{\gamma}_{(3)}$ is no longer a valid diagnosis. If it had checked the same before putting B down, $\vec{\gamma}_{(1)}$ is invalid as well.
- *inserting special "diagnostic actions" in plans and/or*
- *monitoring plan executions*
  in order to detect the necessity of starting a diagnostic routine
- *recovering from error*, i. e.:
  using diagnoses to rectify the performance
- *ontological distinctions between actions*
  E. g., as mentioned above, the robot has control over the initiation of an action, but not over its actual effects: *start_for*, *arrive_at*, and *pick_up* belong to different ontological classes.

In addition to the theoretical investigations we have implemented a prototypical diagnostic system using Prolog — with promising results. At present, called with the example history and observation the system outputs the following diagnoses:

```
history: [pu(B), sf(R), aa(R), pd(B)]
observation: not(at(B, R))
diagnoses:

[pu(B), sf(R), aa(r), pd(B)]
[pu(B), sf(R), aa(R), pd'(B)]
[pu(B), sf(R), aa(r), pd'(B)]
[pu'(B), sf(R), aa(R), pd'(B)]
[pu(B), rl(B), sf(R), aa(R), pd'(B)]
[pu(B), sf(R), rl(B), aa(R), pd'(B)]
[pu(B), sf(R), aa(R), pd(B), st(B)]
[pu'(B), sf(R), aa(r), pd'(B)]
...
```

The first, fourth, sixth, and seventh computed diagnosis correspond to $\vec{\mu}$ (covering $\vec{\gamma}_{(2)}$), $\vec{\gamma}_{(3)}$, $\vec{\gamma}_{(1)}$, $\vec{\gamma}_{(4)}$, respectively.

References

[LPR98]
Hector Levesque, Fiora Pirri, Ray Reiter. Foundations for the Situation Calculus. *Linköping Electronic Articles in Computer and Information Science*, Vol. 3(1998): nr 18, 1998. http://www.ep.liu.se/ea/cis/1998/018/

[McI98]
Sheila A. McIlraith. Explanatory Diagnosis: Conjecturing Actions to Explain Observations. *Proceedings of the Sixth International Conference on Principles of Knowledge Representation and Reasoning (KR-98)*, 1998.

[1] The non-occurrence of an event/action can be represented through a special dummy event/action as a variation.

[2] A diagnosis should be as simple as possible.