Detecting and Repairing Anomalous Evolutions in Noisy Environments: Logic Programming Formalization and Complexity Results

Fabrizio Angiulli[1], Gianluigi Greco[2], and Luigi Palopoli[3]

[1] ICAR-CNR, Via P. Bucci 41C, 87030 Rende, Italy
angiulli@icar.cnr.it
[2] Dept. of Mathematics - Univ. della Calabria, Via P. Bucci 30B, 87030 Rende, Italy
ggreco@mat.unical.it
[3] DEIS - Univ. della Calabria, Via P. Bucci 41C, 87030 Rende, Italy
palopoli@deis.unical.it

Summary. In systems where agents are required to interact with a partially known and dynamic world, sensors can be used to obtain further knowledge about the environment. However, sensors may be unreliable, that is, they may deliver wrong information (due, e.g., to hardware or software malfunctioning) and, consequently, they may cause agents to take wrong decisions, which is a scenario that should be avoided. The paper considers the problem of reasoning in noisy environments in a setting where no (either certain or probabilistic) data is available in advance about the reliability of sensors. Therefore, assuming that each agent is equipped with a background theory (in our setting, an extended logic program) encoding its general knowledge about the world, we define a concept of detecting an anomaly perceived in sensor data and the related concept of agent recovering to a coherent status of information. In this context, the complexities of various anomaly detection and anomaly recovery problems are studied.

1 Introduction

Consider an agent operating in a dynamic environment according to an internal background theory (the agent's trustable knowledge) which is enriched, over time, through sensing the environment. Were sensors completely reliable, in a fully observable environment, the agent could gain a perfectly correct perception of environment evolution. However, in general, sensors maybe unreliable, in that they may deliver erroneous observations to the agent. Thus, the agent's perception about environment evolution might be erroneous and this, in turn, might cause that wrong decisions are taken.

In order to deal with the uncertainty that arises from noisy sensors, probabilistic approaches have been proposed see, e.g., [5, 6, 7, 14, 16, 21]) where evolutions are represented by means of dynamic systems in which transitions among possible states

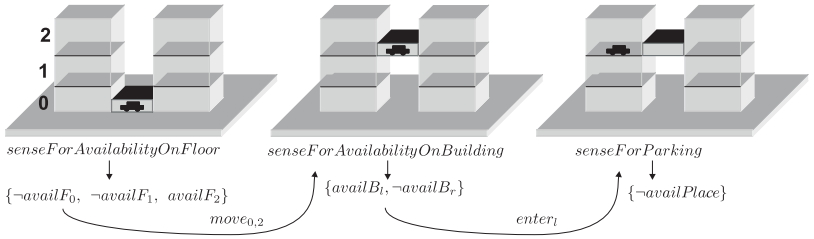

Fig. 1. Parking lot example.

are determined in terms of probability distributions. Other approaches refer to some *logic* formalization (see, e.g., modal logics, action languages, logic programming, and situation calculus [2, 11, 12, 20]) in which a logical theory is augmented to deal quantitatively and/or qualitatively with the reliability of the sensors.

In this paper we take a different perspective instead, by assuming that no information about reliabilities of sensors is available in advance. Therefore, in this context, neither probabilistic nor qualitative information can be exploited for reasoning with sensing. Nonetheless, it is in any case relevant to single out faulty sensor data in order for the agent to be able to correctly maintain a correct perception about the status of the world. To this aim, we introduce a formal framework good for reasoning about anomalies in agent's perception of environment evolutions, that relies on the identification of possible discrepancies between the observations gained through sensors and the internal trustable knowledge of the agent.

In order to make the framework clearer, we next introduce a running example.

1.1 Example of Faulty Sensors Identification

Consider an agent who is in charge of parking cars in a parking lot (see Figure 1). The parking lot consists of two buildings, each with several floors. The floors are reached via a single elevator which runs in the middle in between the two buildings (so, there is a building to *left* and one to the *right* of the elevator door). A number of sensors are used to inform the agent about parking place availability at different levels of the two buildings. In particular, the sensors tell the agent: (a) if there is any available parking place at some level in any of the two buildings (sensor s_1); (b) given the floor where the agent is currently located, if there is any available parking place in the left and/or the right building at that floor (sensor s_2); (c) given the floor and the building (left or right) where the agent is currently located, whether parking places are available at that floor in that building (sensor s_3) – let us assume that there are a total of n parking places at each level of each of the two buildings. Also, the agent uses a background theory that tells him that if he is at floor i of the building x and sensors s_1, when queried, signalled parking availability at level i and sensor s_2, when queried, signalled a parking availability in building x then there must be indeed at least one parking place available at his current position.

Now, assume that, in fact, the agent senses sensor s_3 and the sensor returns the information that no place is available at the current agent's position. This clearly disagrees with the internal state of the agent that tells that there should be indeed

at least one place available in that position. Such disagreement implies that some anomalies came into play somehow.

In particular, the agent might doubt about the reliability of sensor s_3 (that is, there actually are available parking places at the agent position, but s_3 tells that none is available). Similarly, the agent might suspect that sensor s_1 is reliable while s_2 is not, thereby inferring that there is a place to park at the very floor where he is currently located, but on the opposite building.

1.2 Contribution and Organization

Within the framework outlined above, the contribution of the paper is as follows.

In Section 2, we introduce some preliminaries on extended logic programming which shall constitute the basic formalism exploited for modelling agents background knowledge. Then, in Section 3, we formally propose a concept of anomaly in state evolutions of a dynamic environment, as perceived by an agent sensing that environment through (possibly) noisy sensors. Moreover, we define a suitable concept of recovering the agent internal mental state from anomalies.

After that the framework has been introduced, we turn to the study of the computational complexity of some basic relevant problems related to state evolution anomaly detection and recovery. The results are proved and discussed in Section 4. We considered background knowledge bases modelled by means of not-free extended logic programs as well as general logic programs under both the brave and the cautious semantics. We point out here that, depending on the complexity of the agent background knowledge, anomaly checking may be characterized by a quite varied degree of difficulty, ranging from simple checking for the occurrence of complementary literals in sensor data and in the agent background knowledge (which is basically the case for our running example above) to quite complex tasks. The capability of characterizing computational complexity sources in knowledge representation frameworks is important both for gaining knowledge of the structure of the problems the framework comprises and, above all, to be able to realize effective rewriting and optimizations needed to efficiently implement them [10]. This justifies our interest in analyzing the complexity of anomaly detection and repair in agent evolutions, which will accounted for in the paper.

We believe that our investigation is a step towards providing capabilities for dynamic plan monitoring and repairing in noisy environment, where it can be useful for an agent that is trying to achieve its goals to be able to monitor, identify anomalies and fix a plan while evolving [3, 4, 8]. In this respect, it deserves of further work the possibility of prototypically implementing anomalies identification primitives for agent evolutions on top of some available answer set engine (e.g., [13, 17]), and subsequently made them available to conditional planning environments (e.g., [15, 19, 22]).

2 Preliminaries on Extended Logic Programs (ELPs)

We briefly recall here that a propositional ELP is a set of rules of the form $L_0 \leftarrow L_1, \ldots, L_m, \ not \ L_{m+1}, \ldots, not \ L_n \ (n \geq m \geq 0)$, where the symbol "$not$" denotes negation by default, and each L_i is a literal, i.e. an expression of the form p or $\neg p$ with p a propositional letter and the symbol "$\neg$" denotes classical negation. By $\mathbf{h}(r)$ we denote the head L_0 of the rule r, and by $\mathbf{b}(r)$ its body $L_1, \ldots, L_m, not \ L_{m+1}, \ldots, not \ L_n$. An ELP is *positive* if classical negation does not occur in the program.

In the following, we consider the *answer set* semantics for ELPs [9]. *Answer sets* of an ELP P are defined as follows. Let $Lit(P)$ denote the set of all the literals obtained using the propositional letters occurring in P. Let a *context* be any subset of $Lit(P)$. Let P be a *negation-by-default-free* ELP. Call a context S *closed under* P iff for each rule $L_0 \leftarrow L_1, \ldots, L_m$ in P, if $L_1, \ldots, L_m \in S$, then $L_0 \in S$. An *answer set* of P is any minimal context S such that (1) S is closed under P and (2) if S is inconsistent, that is if there exists a propositional letter p such that both $p \in S$ and $\neg p \in S$, then $S = Lit(P)$. An answer set of a general ELP is defined as follows. Let the *reduct of P w.r.t the context S*, denoted by $Red(P, S)$, be the ELP obtained from P by deleting (i) each rule that has $not \ L$ in its body for some $L \in S$, and (ii) all subformulae of the form $not \ L$ of the bodies of the remaining rules. Any context S which is an answer set of $Red(P, S)$ is an *answer set* of P. By $\mathrm{ANSW}(P)$ we denote the collection of all consistents answer sets of an ELP P. An ELP P is ANSW-consistent iff $\mathrm{ANSW}(P) \neq \emptyset$.

An ELP P *cautiously* entails a literal l, written $P \models_c l$, iff for each $S \in \mathrm{ANSW}(P)$, $l \in S$. An ELP P *bravely* entails a literal l, written $P \models_b l$, iff there exists $S \in \mathrm{ANSW}(P)$ such that $l \in S$.

3 Formal Framework

In this section, we introduce a simple framework to model environment state evolutions with sensing, and we formally state the problem of reasoning about possibly faulty sensors. In this respect, we present some techniques that an agent might exploit to identify 'anomalous' observations (and, hence, faulty sensors), and a 'repair approach' in execution monitoring accommodating the uncertainty on the outcome of the sensors.

3.1 Sensors, Agents, and Transitions

Let $\mathcal{F}$ be a set of propositional variables. We denote by $\neg f$ the negation of any $f \in \mathcal{F}$, and by $\mathcal{F}^{\neg}$ the set $\{\neg f \mid f \in \mathcal{F}\}$.

We distinguish two (disjoint) sets of variables: (i) *beliefs B*, denoting the agent beliefs about the status of the world; (ii) *observables O*, modelling the actual status of the world as returned by a set $\mathcal{S}$ of environment sensors. Specifically, for each sensor $s \in \mathcal{S}$, $\lambda(s) \subseteq \mathcal{O}$ denotes the set of propositional variables that are sensed by

$$\mathcal{B}: \begin{cases} floor_0,\ floor_1,\ floor_2 \\ building_l,\ building_r \end{cases} \qquad \mathcal{S}: \begin{cases} \lambda(s_1) = \{availF_0,\ availF_1,\ availF_2\} \\ \lambda(s_2) = \{availB_l,\ availB_r\} \\ \lambda(s_3) = \{availPlace\} \end{cases}$$

$$\mathcal{O}: \begin{cases} availF_0,\ availF_1,\ availF_2 \\ availB_l,\ availB_r \\ availPlace \end{cases} \qquad \mathcal{K}: \begin{cases} availPlace \leftarrow availF_i,\ availB_j, \\ \qquad\qquad\qquad floor_i,\ building_j. \end{cases}$$

$$\begin{cases} senseForAvailabilityOnFloor: & \langle floor_0,\ s_1\rangle \\ senseForAvailabilityOnBuilding: & \langle \emptyset,\ s_2\rangle \\ senseForParking: & \langle \emptyset,\ s_3\rangle \\ move_{i,j}: & \langle floor_i, \{\neg floor_i,\ floor_j\}\rangle \\ enter_j: & \langle \emptyset, \{building_j\}\rangle \end{cases}$$

Fig. 2. Formalization of the parking lot example.

s. Moreover, at any time instant, we assume that the value of the sensor is returned by a function $val : \mathcal{S} \mapsto \lambda(S) \times \lambda(S)^{\neg}$ producing a consistent set of observables (literals which the sensor evaluates to 'true').

Example 1. In the *parking lot* application, the sensors $\mathcal{S}$, the observables $\mathcal{O}$ and the beliefs $\mathcal{B}$ are reported in Figure 2, where, for instance, $availF_i$ means that there are parking places available at level i, $availB_l$ that at the current floor there are places available in the left building, and $building_r$ that the car is currently in the building on the right. $\lhd$

Each agent is characterized by a *background knowledge* $\mathcal{K}$ expressed as an extended logic program over $\mathcal{F}$, and, over time, by a current *state* represented as a pair of sets $S = \langle S_B, S_O\rangle$, where $S_B \subseteq B \times B^{\neg}$ and $S_O \subseteq O \times O^{\neg}$, such that both S_B and S_O are consistent. In the following, S_B (resp. S_O) will be denoted by $\mathcal{B}(S)$ (resp. $\mathcal{O}(S)$). In order to achieve its goals, the agent operates by executing *operators* that cause *transitions* between states, that is, they cause the environment to evolve together with the agent mental state. In particular, a transition usually changes the agent's beliefs; yet, it may change the observables if the corresponding operator includes a *sensing*.

Several ways to define transitions between states in the presence of sensing actions have been proposed in the literature, accounting, e.g., for non-deterministic effects, causal effects, probabilities, and so on (see, e.g., [20, 18, 14, 12, 21] and references therein). In the foregoing, we decided to refer to a particularly simple approach, since the results we are going to present are largely independent of the chosen formalization of transitions and associated operators. Hence, in order to make the exposition clearer we resume to a quite simple model which allows to specify preconditions, multiple-effects and *sensing actions*. So, in our context, an operator t for an agent A is simply a pair $\langle c, e\rangle$ such that c is a logic formula over the set $B \cup O$ denoting the preconditions, and e is the effect of the operator which can be either *(i)* a consistent subset e_B of $B \times B^{\neg}$, or *(ii)* a sensor $e_s \in \mathcal{S}$. We denote by $con(t)$ the precondition c, and by $\mathit{eff}(t)$ the effect e of t. See Figure 2 for the theory $\mathcal{K}$ and the set of operators in our parking lot application.

In the following, given a precondition c and an answer set S we will assume that the entailment $S \models c$ is polynomial time decidable. An ELP P *cautiously* entails a condition c, written $P \models_c c$, iff for each $S \in \mathrm{ANSW}(P)$, $S \models c$. An ELP P *bravely* entails a condition c, written $P \models_b c$, iff there exists $S \in \mathrm{ANSW}(P)$ such that $S \models c$.

We next define the semantics of applying operators to agent's states.

Definition 1. Let A be an agent and $\langle S_B, S_O\rangle$ be a state for it. An operator $t = \langle c, e\rangle$ is *applicable* in $\langle S_B, S_O\rangle$ if $(\mathcal{K} \cup S_B \cup S_O) \models c$, and the *result of its application* is the state $\langle S'_B, S'_O\rangle$ defined as:

- $\langle S_B, (S_O \setminus v^{\neg}) \cup v\rangle$, with $v = val(e_s)$, if $e = e_s \in \mathcal{S}$, and
- $\langle S_B \setminus e_B^{\neg} \cup e_B, S_O\rangle$, if $e = e_B \subseteq B \times B^{\neg}$.

In the case above, we also write $\langle S_B, S_O\rangle \rightarrow_t \langle S'_B, S'_O\rangle$. □

Example 2. Consider again Figure 2, and in particular, the set of operators reported in the bottom part of it: Given the state $\langle\{floor_0\}, \emptyset\rangle$, we can easily see that the operator $move_{1,2}$ is not applicable. Conversely, the agent might apply the operator $senseForAvailabilityOnFloor$ and a possible outcome is $\langle\{floor_0\}, \{\neg availF_0, \neg availF_1, availF_2\}\rangle$. Figure 1 shows an example of transitions between states exploiting such operators. ◁

3.2 Reasoning on Evolutions

The repeated application of operators define an evolution for the agent. Formally, an evolution H for A is a succession of states of the form $\langle S_B^0, S_O^0\rangle \rightarrow_{t_1} \langle S_B^1, S_O^1\rangle \rightarrow_{t_2} \ldots \rightarrow_{t_n} \langle S_B^n, S_O^n\rangle$, such that (*i*) each transition t_i is applicable in the state $\langle S_B^{i-1}, S_O^{i-1}\rangle$ and (*ii*) each state $\langle S_B^i, S_O^i\rangle$ is the result of the application of t_i in $\langle S_B^{i-1}, S_O^{i-1}\rangle$. Intuitively, H represents an actual plan that the agent is performing in order to achieve a given goal starting from the initial state $\langle S_B^0, S_O^0\rangle$.

In the following, $len(H)$ denotes the number of transitions occurring in the evolution H; $state_i(H)$ denotes the ith state of the evolution H; $state(H)$ denotes $state_{len(H)}(H)$; $tr_i(H)$ denotes the ith transition occurred in the evolution; $H[i]$ denotes the evolution $state_0(H) \rightarrow_{tr_1(H)} \ldots \rightarrow_{tr_i(H)} state_i(H)$.

As previously pointed out, while dealing with noisy sensors, there might be evolutions in which the agent finds some discrepancies between its mental beliefs (plus its trustable knowledge) and the observations at hand. The following definition formalizes such a notion of 'disagreement'.

Definition 2. Let H be an evolution for the agent A with knowledge $\mathcal{K}$. A set of observations $W \subseteq \mathcal{O}(state(H))$ is an *anomaly* for A in H if $\forall w \in W$, $th(A, H) \setminus W \models \neg w$, where $th(A, H)$ denotes the theory $\mathcal{K} \cup \mathcal{B}(state(H)) \cup \mathcal{O}(state(H))$. □

Example 3. Let H be the evolution: $t_1 : senseForAvailabilityOnFloor$; $t_2 : move_{0,2}$; $t_3 : senseForAvailabilityOnBuilding$; $t_4 : enter_l$; and $t_5 : senseForParking$.

Assume, now, that sensed values are such that $\mathcal{O}(state_1(H)) \supseteq \{\neg availF_0, \neg availF_1, availF_2\}$, $\mathcal{O}(state_3(H)) \supseteq \{availB_l, \neg availB_r\}$, and $\mathcal{O}(state_5(H)) \supseteq \{\neg availPlace\}$ – see, again, Figure 2. Intuitively, the agent is planning to park at the second floor in the left building after sensing s_1 and s_2. But, the result of sensing s_3 is anomalous, as it disagrees with its mental beliefs (in $\mathcal{K}$) according to which $availPlace$ should be true there. ◁

The agent employed in our running example has a unique possible view of the world, being its knowledge a positive program. In general, an agent may have several possible worlds. Thus, in the following we will distinguish between the cautious and the brave semantics. In particular, while anomaly existence under the cautious semantics expresses that no possible world is consistent with the sensor readings, under the brave semantics a set of sensor readings is anomalous if each sensor reading is inconsistent with some possible world in which all sensors of the set are simultaneously kept quiet.

Given an anomaly, we are interested in finding possible fixes for it, i.e., "alternative" evolutions defined over the same set of transitions in which, however, the result of the sensing actions may differ from the evolution in which the anomaly has been singled out. This is formalized next with the notion of *repair* for an evolution.

Definition 3. An evolution H' for A is a *repair* for H w.r.t. an anomaly W if:
1. $len(H) = len(H')$,
2. $tr_i(H) = tr_i(H')$, for each $1 \leq i \leq len(H)$, and
3. $\forall w \in W \cap \mathcal{O}(state(H'))$, $th(A, H') \setminus W \not\models \neg w$.

Moreover, H' is *non trivial* if $W \cap \mathcal{O}(state(H'))$ is not empty. □

Example 4. For instance, a repair for our running example is obtained by replacing the value returned by sensor s_2 with $\{\neg availB_l, availB_r\}$ while keeping the values returned by s_1 and s_3. This represents the scenario in which the available place is in the opposite building of the same floor. ◁

4 Reasoning with Noisy Sensors

Now that we have defined our formal framework for anomaly detection and repairing of an agent's mental state evolution, we turn to the problem of defining relevant agent's reasoning tasks. Moreover, as already stated in the Introduction, is it important to pinpoint the computational complexity characterizing such tasks, since this is a fundamental premise to devising effective and optimized implementations of our framework.

Specifically, we shall next consider the following relevant problems:

- ANOMALY-EXISTENCE: Given an agent A and an evolution H for it, does there exist an anomaly W for A in H?
- REPAIR-EXISTENCE: Given an agent A and an anomaly W for A in an evolution H, does there exist a (non trivial) repair H' for H w.r.t. W?

	not -free	cautious	brave
ANOMALY-EXISTENCE	P-c	Σ_2^P-c	NP-c
REPAIR-EXISTENCE	NP-c	Σ_2^P-c	Σ_2^P-c
REPAIR-CHECKING	NP-c	Σ_2^P-c	Σ_2^P-c
ANOMALY&REPAIR-CHECK.	P-c	D^P-c	D^P-c

Fig. 3. Complexity of Basic Problems.

- REPAIR-CHECKING: Given an agent A and evolutions H and H', is H' a repair for H w.r.t. some anomaly W for A?
- ANOMALY&REPAIR-CHECKING: Let A be an agent and H an evolution. Given an evolution H' and a set of observables $W \subseteq \mathcal{O}(state(H))$, is W an anomaly for A in H, and H' a repair for H w.r.t. W?

Complexity results concerning problems defined above are depicted in Figure 3. In the following, the complexity of the ANOMALY-EXISTENCE problem for a particular semantics of general logic programs is investigated.

Let T be a truth assignment of the set $\{x_1, \ldots, x_n\}$ of boolean variables. Then, we denote by Φ the boolean formula $\Phi = C_1 \wedge \ldots \wedge C_m$ in conjunctive normal form, with $C_j = t_{j,1} \vee t_{j,2} \vee t_{j,3}$, where each $t_{j,k}$ is a literal on the set of boolean variables $X = x_1, \ldots, x_n$. Recall that deciding the *satisfiability* of Φ is a well-known NP complete problem.

Theorem 1. ANOMALY-EXISTENCE *for ELPs under brave semantics is* NP*-complete.*

Proof:

(Membership) The problem can be solved by a polynomial time nondeterministic Turing machine that guesses a subset $W \subseteq \mathcal{O}(state(H))$ together with $n = |W|$ contexts $S_1, \ldots, S_n$ of $th(A, H) \setminus W$ such that $\neg w_i \in S_i$, and then checks in polynomial time that each S_i is an answer set of the reduct of $th(A, H) \setminus W$ w.r.t. S_i and, hence, of $th(A, H) \setminus W$.

(Hardness) Given the boolean formula Φ, consider the set of observables $\mathcal{O}(\Phi) = \{x_0, x_1, \ldots, x_n\}$, the sensor $s(\Phi)$ with $\lambda(s(\Phi)) = \mathcal{O}(\Phi)$, and the agent $A(\Phi)$ with knowledge $\mathcal{K}(\Phi)$:

$$
\begin{aligned}
r_0 &: sat \leftarrow c_1, \ldots, c_m. \\
r_{1,j} &: c_j \leftarrow \sigma(t_{j,1}). \qquad (1 \leq j \leq m) \\
r_{2,j} &: c_j \leftarrow \sigma(t_{j,2}). \qquad (1 \leq j \leq m) \\
r_{3,j} &: c_j \leftarrow \sigma(t_{j,3}). \qquad (1 \leq j \leq m) \\
r_{4,i} &: \neg x_i \leftarrow not\ x_i, sat. \ (0 \leq i \leq n)
\end{aligned}
$$

where $\sigma(x_i) = x_i$ and $\sigma(\neg x_i) = not\ x_i$, the operator $t(\Phi) = \langle \emptyset, s(\Phi) \rangle$, and the evolution $H(\Phi) = \langle \emptyset, \emptyset \rangle \rightarrow_{t(\Phi)} \langle \emptyset, \mathcal{O}(\Phi) \rangle$. Now we prove that there exists an anomaly $W \subseteq \mathcal{O}(\Phi)$ for $A(\Phi)$ in $H(\Phi)$ iff Φ is satisfiable.

($\Rightarrow$) Assume that there exists an anomaly W for $A(\Phi)$ in $H(\Phi)$. Then $th(A(\Phi), H(\Phi)) \setminus W \models \neg w$, $\forall w \in W$. As the negation of some observable x_i can be implied only by rule $r_{4,i}$, then it is the case that there exists an answer set M of $th(A(\Phi), H(\Phi))$ such that $sat \in M$. Consequently, $T(x_i) = \mathbf{true}$, $\forall x_i \in X \setminus W$, and $T(x_i) = \mathbf{false}$, $\forall x_i \in W$, is a truth assignment to the variables of Φ that makes the formula satisfied.

($\Leftarrow$) Assume that Φ is satisfiable, and let T_X be a truth value assignment to the variables in X that makes Φ true. Then $W = \{x_0\} \cup \{x_i \mid T_X(x_i) = \mathbf{false}\}$ is an anomaly for $A(\Phi)$ in $H(\Phi)$.

According to the theorem above, negation by default makes ANOMALY-EXISTENCE intractable. Moreover, under the cautious semantics the problem is more difficult than under the brave (see Figure 3), unlike most cases in which a kind of symmetry holds between the complexity of the two semantics (within the same level of the polynomial hierarchy).

5 Conclusions

In this paper we have defined a formal framework good for reasoning about agents' mental state evolution about environments sensed through possibly unreliable sensors. In our framework, no information (neither certain nor probabilistic) is assumed to be available in advance about the reliability of sensors. The agent's perception can however be maintained to encode a correct perception of the world through the identification and the resolution of discrepancies occurring between sensor delivered data and the agent's internal trustable knowledge, encoded in the form of an ELP under answer set semantics. After having defined the formal framework, in order to pinpoint main computational complexity sources implied in the implementation of the anomaly detection and repairing agent's mental state evolution, several reasoning problem have been considered and their complexity have been studied.

We note that the problem of *belief change* is only loosely related to work here done. Indeed, rather than being interested in revising the agent theory in order to entail the new information provided by the environment, we are interested in singling out environmental manifestations to be doubted about. The notion of minimal repair is indeed relevant in order to rank different possible repairs. We point out that several relations of preference between repairs can be embedded in the basic framework here introduced. Indeed, a natural form of preference relies in the number of agent observations the repair should change in order to recover its mental consistency, while it is also interesting to rank repairs depending on the number of anomalies they are able to fix. Furthermore, while sensors under consideration can only report binary states, the framework is not limited to the management of binary environmental measures, as many-valued discrete signals can be indeed simulated by sets of binary signals. Investigating the impact of enriching the framework with sensors delivering real-valued data is also of interest. Finally, it is interesting to explore how the presented framework could be embedded within a full-fledged conditional agent planning system. All those issues discusses above will be the topics of future investigation, while we are currently involved with the implementation of our system on top of the DLV system [13].

References

1. F. Angiulli, R. Ben-Eliyahu-Zohary, and L. Palopoli. Outlier detection using default logic. *in Proc of IJCAI'03*, pp. 833-838.
2. F. Bacchus, J.Y. Halpern, and H.J. Levesque. Reasoning about noisy sensors and effectors in the situation calculus. *Artificial Intelligence*, 111(1-2): 171-208, 1999.
3. M. Balduccini and M. Gelfond. Diagnostic reasoning with a-prolog. *TPLP*, 3(4-5): 425-461, 2003.
4. C. Baral, S.A. McIlraith, and T.C. Son. Formulating diagnostic problem solving using an action language with narratives and sensing, *in Proc of KR'02*, pp. 311-322.
5. C. Baral, N. Tran, and L.C. Tuan. Reasoning about actions in a probabilistic setting. *in Proc. of AAAI/IAAI'02*, pp. 507-512.
6. C. Boutilier, R. Dean, and S. Hanks. Planning under uncertainty: structural assumptions and computational leverage. *JAIR*, 11: 1-94, 1999.
7. C. Boutilier, R. Reiter, and B. Price. Symbolic dynamic programming for first-order MDPs. *IJCAI'01*, pp. 690-700.
8. M. Fichtner, A. Großmann, and M, Thielscher. Intelligent execution monitoring in dynamic environments, *Fundamenta Informaticae*, 57(2-4):371-392. 2003.
9. M. Gelfond and V. Lifschitz. Classical Negation in Logic Programs and Disjunctive Databases. *New Generation Comput.*, 9(3-4): 365-386, 1991.
10. G. Gottlob. Complexity and Expressive Power of Disjunctive Logic Programming. *in Proc of ILPS'94*, pp. 453-465.
11. E. Giunchiglia, J. Lee, V. Lifschitz, N. McCain, and H. Turner. Nonmonotonic Causal Theories. *Artificial Intelligence*, 153(1-2): 49-104, 2004.
12. L. Iocchi, T. Lukasiewicz, D. Nardi, and R. Rosati. Reasoning about Actions with Sensing under Qualitative and Probabilistic Uncertainty. *in Proc of ECAI'04*, pp. 818-822.
13. N. Leone, G. Pfeifer, W. Faber, F.Calimeri, T.Dell'Armi, T. Eiter, G. Gottlob, G.Ianni, G.Ielpa, C.Koch, S.Perri, and A.Polleres. The dlv system for knowledge representation and reasoning. *in Proc. of JELIA'02*, pp. 537-540.
14. M.L. Littman, J. Goldsmith, and M. Mundhenk. The Computational Complexity of Probabilistic Planning. *JAIR* 9:1-36, 1998.
15. J. Lobo. COPLAS: a COnditional PLannner with Sensing Actions. *FS-98-02*, AAAI, 1998.
16. S.M. Majercik and M.L. Littman. Contingent planning under uncertainty via stochastic satisfiability. *Artificial Intelligence*, 147(1-2):119-162, 2003.
17. I. Niemelä and P. Simons. Smodels: An implementation of the stable model and well-founded semantics for normal LP. *in Proc. of LPNMR'97*, pp. 420-429.
18. J. Rintanen. Expressive Equivalence of Formalisms for Planning with Sensing. *in Proc. of ICAPS'03*, pp. 185-194.
19. J. Rintanen. Constructing conditional plans by a theorem prover. *JAIR*, 10:323-352, 2000.
20. T.C. Son, P.H. Tu, and C. Baral. Planning with Sensing Actions and Incomplete Information Using Logic Programming. *in Proc of LPNMR'04*, pp. 261-274.
21. T.C. Son and C. Baral. Formalizing sensing actions A transition function based approach. *Artificial Intelligence*, 125(1-2):19-91, 2001.
22. M. Thielscher. Programming of Reasoning and Planning Agents with FLUX. *in Proc. of KR'02*.