Reasoning about Risk in Agent's Deliberation Process: a Jadex Implementation

Yudistira Asnar, Paolo Giorgini, and Nicola Zannone

Department of Information and Communication Technology
University of Trento, Italy
{yudis.asnar,paolo.giorgini,zannone}@dit.unitn.it

Abstract. Autonomous agents and multi-agent systems have been proved to be useful in several safety-critical applications. However, in current agent architectures (particularly BDI architectures) the deliberation process does not include any form of risk analysis. In this paper, we propose guidelines to implement Tropos Goal-Risk reasoning. Our proposal aims at introducing risk reasoning in the deliberation process of a BDI agent so that the overall set of possible plans is evaluated with respect to risk. When the level of risk results too high, agents can consider and introduce additional plans, called treatments, that produce an overall reduction of the risk. Side effects of treatments are also considered as part of the model. To make the discussion more concrete, we illustrate the proposal with a case study on the Unmanned Aerial Vehicle agent.

1 Introduction

Agent technology is becoming more and more an emergent alternative to build safety-critical systems [1, 2]. Humans are replaced by autonomous agents in high risk situations or safety-critical missions, such as reconnaissance and battle-combat. However, developing autonomous agents for such tasks require to introduce and consider risk and related problems within the agent's deliberation process.

Many models dealing with agent's mental states have been proposed in literature [3–5]. Most of them describe agent's mental states in terms of Belief, Desire, and Intention (the BDI model). The BDI model has been initially introduced by Bratman [6] and then refined by Rao and Georgeff [7, 4] for real implementation in agent based systems. Currently, agent tools and frameworks, such as Jack [8], Jadex [9], and Jason [10], use effectively the BDI model in their implementations. Here the deliberation process of an agent is supported by a meta-level reasoning, where the most appropriate plan for achieving a given goal is selected on the basis of specific criteria such as priority and beliefs. Unfortunately, these implementations do not consider uncertain events and, in particular, risks (i.e., uncertain events with negative impacts [11]) as integral part of the meta-level reasoning. On the other hand, several approaches have been proposed in literature to reason about uncertainty [12, 13], but often their complexity made almost impossible the implementation in real agent-based applications [12, 14].

In this paper, we adopt and adapt Tropos Goal-Risk (GR) Framework [15] within the deliberation process of a BDI agent. The GR framework extends the Tropos Goal Model [16] adopting the idea of the three layers analysis introduced by Feather et

al. [17] in their Defect Detection and Prevention (DDP) framework. The GR framework consists of three layers: goal, event/risk, and treatment. These three layers are used to reason about uncertain events that obstruct goals and to evaluate the effectiveness of treatments.We propose an implementation of the GR framework in Jadex [9]. The idea is to encode the GR framework into the Jadex platform so that the deliberation process of an agent takes into account risks and associated costs.

The paper is structured as follows. Section 2 provides a brief description of the Unmanned Aerial Vehicle agent that will be used as an example to explain the whole framework. Section 3 explains the Tropos Goal-Risk framework, and Section 4 details its implementation in Jadex. Finally, Section 5 discusses related work and Section 6 gives final remarks.

2 Unmanned Aerial Vehicle

An Unmanned Aerial Vehicle (UAV) is an aircraft without pilot, that either is controlled remotely or flies autonomously. UAVs are typically used in a number of critical missions, such as decoy, reconnaissance, combat, and even for research and civil purposes. In the early time, UAVs were called drones because they were not more than aircraft remotely controlled by human.

Recently, several efforts have been made to apply intelligent agents to control aircraft [18–20]. Attempts to use intelligent agents in managing UAVs are addressed to avoid the risk of human life loss. In this setting, agents can respond to event occurrences autonomously without waiting for instructions from the ground control. This capability results to be essential in critical and dangerous missions (e.g., reconnaissance, decoy, and combat) where the response time has to be as short as possible. For instance, if a drone understands that it has been detected by the enemy, it informs the ground control about the danger. However, it will still proceed with the mission according to the previous plan until it receives new instructions from the ground control. It is possible that these new instructions are sent to the UAV too late for ensuring the success of the mission. The ambitious objective of the agent paradigm is to provide UAV with facilities to react autonomously and so to take countermeasures in the appropriate time.

There are still open problems for completely replacing human with agents. For instance, a human pilot can learn from past experience so that it can adopt adequate measures when a new occurrence of events is detected. This paper aims at improving software agents with such a capability.

3 Tropos Goal-Risk Model

Tropos is a software engineering methodology that adopts the concept of agent and its related mentalistic notions (e.g., goals, tasks, and resources) along the whole software development process [21]. The Tropos Goal-Risk (GR) framework [15, 22] enhances the Tropos goal model by extending the conceptual model with constructs and relations specific to risk analysis. Basically, a GR model is represented as a graph $\langle \mathcal{N}, \mathcal{R} \rangle$, where $\mathcal{N}$ are nodes and $\mathcal{R}$ are relations. $\mathcal{N}$ is comprised of goals, plans,

and events, and $\mathcal{R}$ consists of decomposition (AND/OR), contribution, and means-end relations. *Goals* are defined as strategic interests that an agent may have, while an *event* is an uncertain circumstance/state of affair that may affect (positively or negatively) the achievement of a goal. Typically, events are out of agents' control. A *plan* is defined abstractly a course of actions, which can be used to achieve a goal or to reduce/treat/mitigate the effects of an event. To distinguish between the plans used to achieve a goal (hereafter plan) and the ones for mitigating the risk, we call mitigating plans *treatments*.

Goals, plans, treatments, and events are characterized by two attributes: satisfaction (SAT) and denial (DEN). SAT represents the supporting evidence one has about the achievement of a goal, the execution of a plan, or the occurrence of an event. DEN represents the evidence about the failure in fulfilling a goal, executing a task, or the occurrence of an event. Though they have similar intuition with probability theory, SAT and DEN are not related and cannot be derived one from the other. The values of attributes are qualitatively represented as $\{F\}$*ull*, $\{P\}$*artial*, and $\{N\}$*one*, with intended meaning $F > P > N$. For instance, $Sat(G) = P$ means that there is (at least) partial evidence that goal G will be achieved, whereas $Den(G) = N$ means that there is no evidence about the failure in achieving goal G. For plans and treatments, these attributes are called *success-rate* and represent how likely a plan or a treatment will be successfully executed. Besides success-rate, plans and treatments are also associated with the cost for their execution. In the UAV scenario, for instance, it may refer to the power consumed to execute a plan. SAT and DEN are also used to represent the likelihood of an event, following the idea of the Theory of Evidence proposed by Dempster and Shaffer [23]. In a GR model $\langle \mathcal{N}, \mathcal{R} \rangle$, relations in $\mathcal{R}$ are represented as $(N_1, \ldots, N_n) \overset{r}{\longmapsto} N$, where r is the type of the relation, $N_1, \ldots, N_n$ are called *source nodes* and N is the *target node*. The formal framework adopts the axioms for the semantic of relations from [24]. This framework is presented in [22] along the rules used to propagate evidence from source nodes to the target node.

A GR model is composed of three different layers: goal layer, event layer, and treatment layer. The **goal layer** models the strategic interests of an agent (i.e., goals) and corresponding plans to obtain them. For instance, Fig. 1 shows the GR model of the UAV agent (goals are represented as ovals and plans as hexagonal). The UAV agent has as main goal to investigate the enemy area (G_1) which is AND-decomposed into define the flight airways to the target (G_4), fly to the target (G_5) based on the defined airways, identify the target (G_6) whether it is a relevant target or just a decoy, and take the picture of target (G_7). According to the rules, which is adopted from [24], the SAT value for G_1 is calculated as the minimum of all SAT values of subgoals G_4, G_5, G_6, and G_7. Inversely, the DEN value for a target goal is calculated as the maximum of all DEN values of its subgoals. Differently, for OR-decomposition the achievement of a subgoal implies the achievement of its root goal. For instance, to achieve goal take the picture of target (G_7), the UAV agent can use take-store schema (G_8) or use take-transmit schema (G_9). The refinement process terminates when all leaf goals are tangible, i.e., for each leaf goal there is at least a plan that can be used to achieve it. For instance, to achieve goal fly to the target (G_5), the UAV can adopt two alternative plans: fly at high altitude (P_3) or fly at low altitude (P_4). These

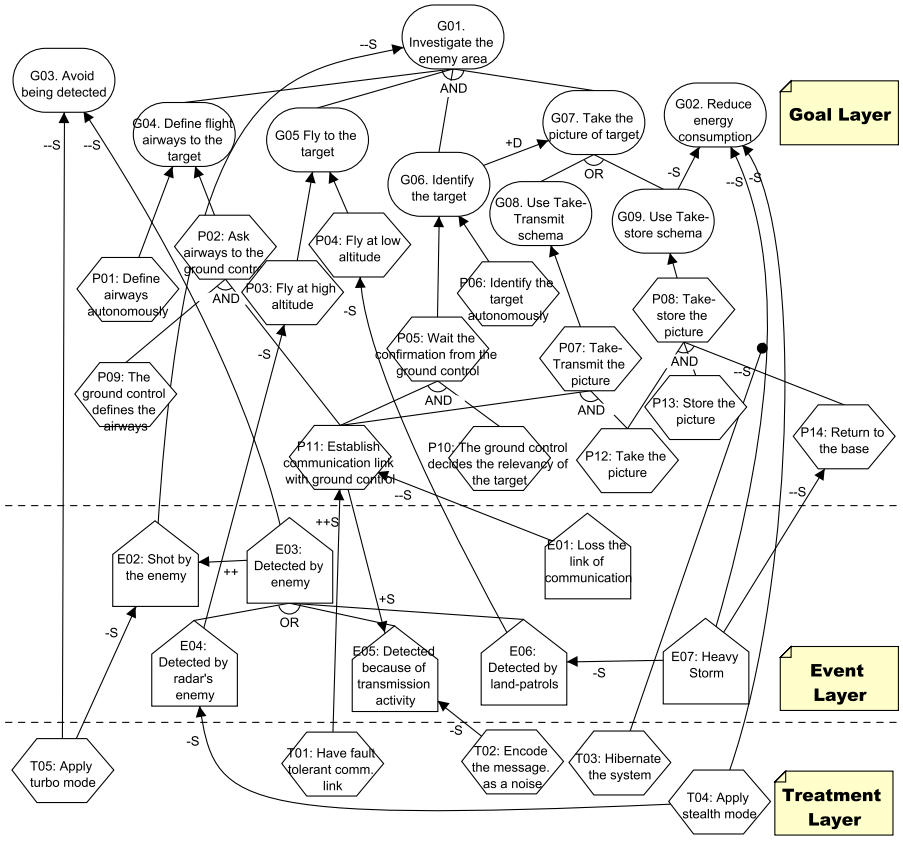

Fig. 1. Goal-Risk model of the UAV Agent

plans are connected by means-end relation to their end G_5 (indicated as $P_3 \longmapsto G_5$ and $P_4 \longmapsto G_5$, respectively). Each plan may further be refined using decomposition relations.

Additionally, there are situations where the achievement of a goal can contribute positively or negatively (denoted by $+$,$++$,$-$,$--$) to the achievement of other goals. In our example, the goal use take-transmit schema (G_9) requires more energy and so it negatively contributes $(-_S)$ to the satisfaction of goal reduce the energy consumption (G_2) (indicated as $G_9 \stackrel{-_S}{\longmapsto} G_2$). The contribution $-_S$ states that the SAT value of the source goal is propagated to the DEN value of the target goal with the maximum value $Partial$. $--_S$ has a similar semantics but they can contribute till *Full*. Negative contribution relations for denial ($--_D$ and $-_D$) have dual propagation rules. Similar considerations can be applied to positive contribution. For instance, the denial of goal identify the target (G_6) contribute positively $(+_D)$ to the denial of goal take the picture of target (G_7) since the UAV has to decide whether or not the target is relevant before taking the picture. Similar rules apply for positive contribution relations

for satisfaction ($++_S$ and $+_S$). Finally, there could be situations where both $+_S$ and $+_D$ are required for the same goals. In this case, we simply use $G_x \overset{+}{\longmapsto} G_y$ which means that both $G_x \overset{+_S}{\longmapsto} G_y$ and $G_x \overset{+_D}{\longmapsto} G_y$ hold.

In the **event layer** uncertain events (depicted as pentagons) are analyzed along their influence to the goal layer using contribution relations (Fig. 1). Notice that the occurrence of events can affect the fulfillment of goals and the execution of plans, but their absence do not affect them. This is because the occurrence of an event (e.g., heavy storm (E_7)) delivers negative evidence to the goal layer (especially the goal reduce energy consumption (G_2)), while the absence of the event does not deliver any evidence neither positive nor negative. Based on this observation, we assume that contribution relations from the event layer to other layers propagate only SAT evidences (denoted by $++_S$, $+_S$, $--_S$, and $-_S$). Though in this paper we are particularly interested to the negative effect of event (risk), the GR model also handles events with positive impacts over goals.[1] For instance, the event heavy storm (E_7) can cause an increment of the fuel consumption, that negatively contributes $(-)$ to the achievement of goal reduce energy consumption (G_2). At the same time, E_7 reduces the likelihood of being detected by land-patrols (E_8). As in the Probabilistic Risk Analysis (PRA) [25], we characterize an event with two attributes: *likelihood* of its occurrence and *severity* once it occurs. As mention before, likelihood is realized as SAT and DEN values representing the evidence that support or prevent an event to be occurred. Severity is represented as the sign of a contribution relation (e.g., $+$,$++$,$-$,$--$). As in Fault Tree Analysis [26], an event can be analyzed using decomposition relations. The formal semantics of these relations are the same with the one in the goal layer.

Typically, events are out of an agent's control and the only thing that the agent can do is to try mitigating their impacts. An agent can adopt specific treatments to either reduce the likelihood or reduce the severity of risk. In the **treatment layer** specific plans are introduced and related to the goal and event layer. As shown in Fig. 1, goal apply stealth mode (T_4) technology reduces the likelihood of risk being detected by enemy's radar (E_4). This relation is depicted as $T_4 \overset{-}{\longmapsto} E_4$. In case of storm, the UAV agent can hibernate the system (T_3) to reduce the negative impacts of the event E_7 towards the achievement of goal reduce the energy consumption (G_2). This relation is depicted as a line ending with a black circle in Fig. 1 and represented as $T_3 \overset{--}{\longmapsto} (E_7 \overset{--}{\longmapsto} G_2)$. We call this relation *alleviation relation*, but for sake of simplicity, we do not detail it in this paper. The level of risk effect reduction from a treatment depends on the success-rate of the treatment and the sign of the relation to the event layer.

The overall GR model represents exactly what an agent can do to achieve its goals (i.e., applying the right plans) and to mitigate associated risks (i.e., adopting necessary treatments). To choose among the possible plans and treatments, an agent needs to reason on its GR model, and later the combination of plans and treatments is called as strategy. To this end, we proposed a risk analysis process (or risk reasoning) [22] to select appropriate plans and treatments, such that the risk is acceptable and the cost (i.e., power consumption) is affordable. Suppose an UAV agent must be operated within the

[1] Notice that a negative event for an agent, may be positive for another agent.

maximum risk level (called RISK) and the affordable cost (called COST) that is needed to proceed its strategy. In particular, the process synthesizes a strategy (i.e., plans and treatments) from GR model $\langle \mathcal{N}, \mathcal{R} \rangle$such that the agent can satisfies its top goals (e.g., G_1, G_2, and G_3) with the risk below RISK and the total cost of the strategy below COST. The risk of this strategy is calculated using forward reasoning [24] with taking the likelihood of events and the success-rate of plans and treatments, in terms of SAT and DEN values, as inputs, and the SAT and DEN values of the top goals must be below the RISK. The total cost is defined as the sum of the cost to execute selected plans and to employ necessary treatments.

Initially, the agent generates all possible solutions (i.e., sets of leaf subgoals) for achieving top goals in $\langle \mathcal{N}, \mathcal{R} \rangle$. For instance, the UAV agent has top goal G_1 that can be fulfilled by fulfilling two sets of input-goals, $\{G_4, G_5, G_6, G_8\}$ or $\{G_4, G_5, G_6, G_9\}$ (Fig. 1). Goal models are AND/OR trees and the problem of finding all the possible combinations of subgoals that satisfy top goals corresponds to a boolean satisfiability problem [27]. Our approach is based on the solution proposed in [28], where Tropos goal models are encoded into satisfiability formulas and SAT solvers are then used to find input-goals (i.e., goals that must be satisfied) that satisfy the top goals. For a given set of input-goals, the agent finds all possible set-of-plans that are means for the input-goals (i.e., a plan that is connected by means-end relation to a goal). For instance, $\{P_1, P_3, P_6, P_{11}, P_{12}\}$ is one of the possible set-of-plans that satisfies the top goal G_1 (Fig. 1).

Once the cost of the set-of-plans is affordable (i.e., less than COST), the agent continues to assess the risk of the set-of-plans. If the risk is also acceptable (i.e., less than RISK), the set-of-plans is defined as the strategy to be executed by the agent. Otherwise, if the risk is unacceptable (i.e., higher than RISK), the agent must find applicable treatments to mitigate risk such that the risk is acceptable. For instance, the set-of-treatments $\{T_4, T_5\}$ is one of the possible measures that the UAV can adopt to reduce the risk to be detected by the enemy. Before adding the set-of-treatments as part of the strategy, the agent needs to verify whether the total cost and risk are still affordable. If they are acceptable, the set-of-treatment and the set-of plans are considered as the strategy of the agent, otherwise the agent restart to analyze other set-of-plans.

At the end, the risk assessment process returns a strategy (i.e., plans and treatments) used to fulfill top goals considering the level of risk and the total cost associated with them. In this paper, we assume that there is always a solution for each attempt finding a strategy.

4 Framework Realization

Several agent infrastructures based on BDI concepts, such as Jack [8], Jadex [9], and Jason [10], have been proposed in the last years. Typically, agent platforms follow the Reactive Reasoning and Planning (PRS) computational model [29] to implement their reasoning engine. PRS-like systems are event-based where events[2] are used to denote incoming messages, a new goal to be processed, or a change in the state of an existing goal. Once an event is generated, the agent dispatches the event to the deliberation

[2] The notion of event supported by agent platforms is a different from the one in Tropos.

mechanism (called meta-level reasoning) to choose the most appropriate plan to handle it. In some PRS-system, like Jack and Jason, goals are represented as special type of event, called goal-event. Thus, agents implemented in Jack or Jason platform do not know the current pursuing goals. They execute the plan only as a response to the occurrence of an event.

On the contrary, the notion of goal plays a key role in the Jadex platform. This has spurred us to choose Jadex as platform to implement the GR framework. Jadex requires to specify agents' goals explicitly. These goals must be related to the plans that provide means to achieve them. Jadex represents agents' beliefs as Java objects and stores them in the beliefbase, that is, the database of agents' beliefs. Moreover, Jadex allows one to specify the plan that has to be executed when a belief is changed. Those BDI descriptions are represented in an XML file, called *Agent Definition File* (ADF) by a Jadex agent. The Jadex reasoning engine starts the deliberation process by considering the goals requested by the agent. To this end, it adopts the goals stored in the database that contains all adopted goals by the agent, called the agent's goalbase. To distinguish between just adopted and actively pursued goals, Jadex distinguishes a goal into three states: option, active, and suspended. When a goal is adopted, it becomes an option goal in the agent's goalbase. Once the agent pursues the goal, it becomes active and a goal-event is triggered. The event is then dispatched to the meta-level reasoner in order to find the most appropriate plan to achieve the goal.

The process to implement the GR models of the UAV agent into the Jadex platform starts by defining its beliefs (Fig. 2(a)). The agent's beliefs include: the thresholds of risk level and cost (line 2-7) that are acceptable for the UAV agent; the current cost and risk level of adopted strategies are encoded in (line 8-10) and (line 11-13) respectively. The success rate for plans, the likelihood of events, and the success rate for treatments, which are depicted as Goal-Risk labels, are also included as the beliefs (line 17-32). We also provide the agent with facilities for risk analysis by representing the GR model as a Java object and storing it as a belief (line 14-16). We also specify assess_risk plan (line 2-9 in Fig. 2(b)) in case there is a change in these belief values. Essentially, assess_risk calculates the risk level and cost of adopted strategy (i.e., plans and treatments) using a Jadex implementation of *Forward Reasoning* [24].

To contain the risk level and cost, the UAV agent needs to introduce two additional goals: maintain_risks (line 2-9 in Fig. 2(c)) below the risk threshold and maintain_costs below the cost threshold. If such goals are denied, plan re-planning (line 10-15 in Fig. 2(b)) is executed to recover the desired conditions. Essentially, re-planning is the Jadex implementation of *Risk Reasoning* and is used to synthesize a strategy on the basis of the current beliefs. As mentioned previously, we assume that there is always a strategy for each re-planning attempt.

We also need to represent goals, plans, and treatments of the GR model in the ADF, so that the Jadex implementation of the agent behaves accordingly to the GR model at execution time. Goals are declared in the ADF goal-base, while plans and treatments are depicted as plans section of ADF. This transformation starts by declaring all goals, plans, and treatments occurring in the GR model as XML entries in the ADF (Fig. 2(c) and Fig. 2(b)). Additional Jadex-goals and Jadex-plans may be introduced to mimic the behaviors supported by the GR framework, that are missing in the Jadex platform.

```
<beliefs>
<belief name="thres_cost"  class="double">
<fact>2000000</fact>
</belief>
<belief name="thres_risk"  class="char">
<fact>P</fact>
</belief>
<belief name="cost"  class="double">
<fact>1000000</fact>
</belief>
<belief name="risk"  class="char">
<fact>N</fact>
</belief>
<belief name="gr_model"  class="GRmodel">
<fact>new GRmodel("uav.grmodel")</fact>
</belief>
<beliefset name="goals" class="GRlabel">
<fact>new GRlabel(G1,N)</fact>
...
</beliefset>
<beliefset name="plans" class="GRlabel">
<fact>new GRlabel(P1,F)</fact>
...
</beliefset>
<beliefset name="events" class="GRlabel">
<fact>new GRlabel(E1,P)</fact>
...
</beliefset>
<beliefset name="treatments" class="GRlabel">
<fact>new GRlabel(T1,N)</fact>
...
</beliefset>
...
</beliefs>
```

(a) Beliefs

```
<plans>
<plan name="assess_risk">
<body>new RiskAssessment()</body>
<trigger>
<beliefsetchange ref="events"/>
<beliefsetchange ref="plans"/>
<beliefsetchange ref="treatments"/>
</trigger>
</plan>
<plan name="replanning">
<body>new Replanning()</body>
<trigger>
<goal ref="maintain_risk"/>
</trigger>
</plan>
<plan name="P-G01">
<body>..</body>
<trigger>
<goal ref="G01"/>
</trigger>
</plan>
<plan name="chooseG07">
<body>new RiskReasoning()</body>
<trigger>
<goal ref="choose_planG07"/>
</trigger>
</plan>
<plan name="P7">
<body>..</body>
<trigger>
<goal ref="G8"/>
</trigger>
</plan>
...
</plans>
```

(b) Plans

```
<goals>
<maintaingoal name="maintain_risk">
<maintaincondition>
\$beliefbase.risk < \$beliefbase.risk.thres_risk
</maintaincondition>
<targetcondition>
\$beliefbase.risk < \$beliefbase.risk.thres_risk
</targetcondition>
</maintaingoal>
<achievegoal name="G01">
...
</achievegoal>
...
<achievegoal name="G07">
...
</achievegoal>
<metagoal name="choose_planG07">
...
<trigger>
<goal ref="G07"/>
</trigger>
</metagoal>
<achievegoal name="G08">
...
</achievegoal>
...
</goals>
```

(c) Goals

Fig. 2. UAV agent description in Jadex-ADF

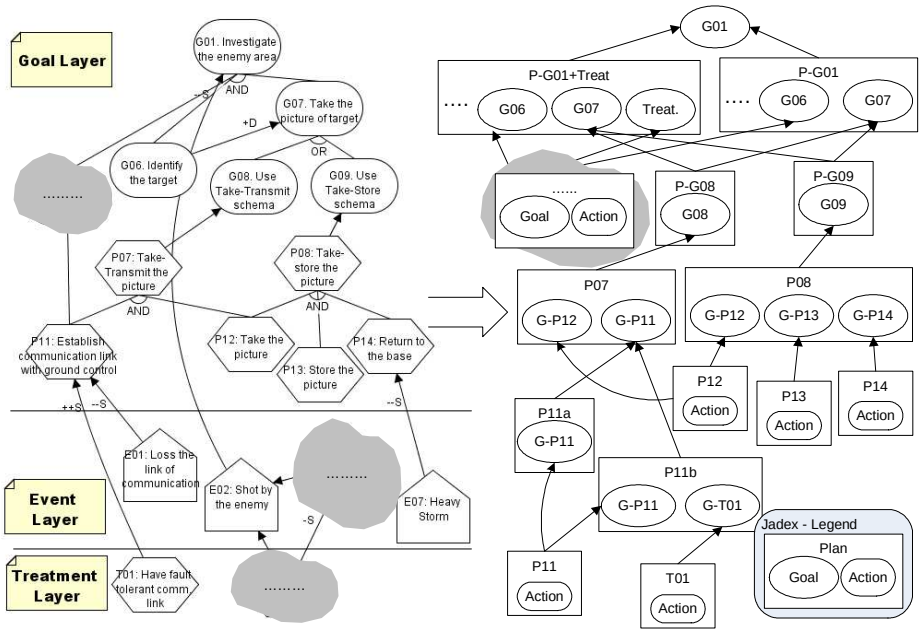

Fig. 3. Goal-Risk model to Jadex

A Jadex-plan can introduce a new goal, instead only performs a course of action (as denoted in Jadex-Legend Fig. 3).

Goals, plans and treatments in Tropos can be AND/OR-decomposed, while in Jadex a goal can only be AND-decomposed. Indeed, Tropos decompositions represent the knowledge of agents about the relations between goals. On the other hand, Jadex introduces subgoals as a result of a plan. For instance, G_1 is AND decomposed into G_6 and G_7 (left-side in Fig. 3). To realize this behavior during execution of a Jadex agent (right-side in Fig. 3), we need to introduce an additional plan P-G01. This plan introduces (and later dispatches) subgoals G_6 and G_7, as depicted in the right-side of Fig. 3. P-G01 is means to achieve G_1 (depicted by arrow line), and the ADF description is shown in Fig. 2(b) (line 16-21). However, both Tropos and Jadex require the fulfillment of all AND-subgoals for having evidence about the fulfillment of upper-level goal. AND-decompositions of plans/treatments have a similar intuition, with introducing additional goals to activate the subplans. For instance, plan P_8 will introduce additional goals (i.e., G-P12, G-P13, and G-P14) that later activate subplans P_{12}, P_{13}, and P_{14}.

To mimic goal OR-decompositions, we introduce additional plans where each plan is used to activate a new subgoal. These plans represent alternative means that an agent can adopt to achieve the upper-level goal. For instance, G_7 is OR-decomposed into G_8 and G_9 (Fig. 3). This is represented in the ADF by introducing the additional plans P-G08 and P-G09 that respectively activate G_8 and G_9, as ways to achieve goal G_7. The agent will perform meta-level reasoning on meta-goal of G_7 (called choose_planG07 in Fig. 2(c) line 17-22) to decide which plan should be adopted. In other words, we over-

ride the Jadex meta-level reasoning by defining a new plan (chooseG07 in Fig. 2(c)) to determine the least risky alternative. Such a plan has body RiskReasoning which is a Java implementation of *Risk Reasoning*. OR-decompositions of plans/treatments can be mimicked by defining a task that introduces an additional goal, and all subplans are defined as means to achieve the additional goal. Moreover, we must override the Jadex meta-level reasoning with a similar plan with chooseG07 which uses RiskReasoning as the body.

Finally, we need to represent Tropos means-end relations in a Jadex agent. Means-end relations correspond to the trigger mechanism in Jadex. For instance, $P_7 \longmapsto G_8$ is represented by adding G_8 as the trigger of P_7 in the ADF (line 30-32 in Fig. 2(b)) or denote by relating P07 to G08 with an arrow line (right-side in Fig. 3). This mapping scheme holds if the plan is not related with any treatments (i.e., direct or indirectly through the event layer). On the contrary, if the plan is related with a treatment, the mapping schema is slightly different because several combinations of plans and treatments can be used to achieve the goal (e.g., it can adopt only the plan, the plan with a treatment, the plan with some treatments, or the plan and all treatments). For instance, the success of plan establish communication with ground control(P_{11}) is obstructed by the risk due to loss the link of communication(E_1) (Fig. 3). In this setting, the UAV agent can adopt treatment have fault tolerant communication link(T_1) for retention purpose. The agent may adopt only P_{11} (or P11a in Fig. 3 (right-side)), whereas in others it needs to adopt P11b, which is the combination of P11 and T1, for ensuring the success of P_{11} execution. This mapping is getting more complicated when several treatments can be applied since the number of combinations of plans and treatments to achieve a goal is exponential $(Numb(plan) \times 2^{Numb(treatment)})$. For instance, if G_1 is obstructed by E_2 directly and by E_3 indirectly and the agent knows a single plan to achieve G_1 and three treatments (e.g., T_2, T_4, and T_5) to mitigate E_2 and E_3, the agent has eight possible strategies to achieve goal G_1.

The proposed Jadex-implementation allows an agent to perform meta-level reasoning using GR framework. In particular, we have taken advantages of combining GR and Jadex reasoning facilities. The Tropos approach explores all alternative solutions and chooses an acceptable (and optimal) sequence of plans and treatments, whose execution allows the agent to achieve the top goals from the current condition. Conversely, Jadex reasoning only elicit a plan to achieve the goal, and later could be the case that the plan will introduce other goals or trigger the execution of other plans. We have implemented *Risk Reasoning* using Jadex approach. Thus, the agent keeps performing *Risk Reasoning* each time there are several applicable plans, though *Risk Reasoning* has already provided a strategy for pursuing a goal. Ideally, the reasoning mechanism is executed in the beginning, and it will be executed again if there is a change in the agent's beliefs to revise the existing strategy.

5 Related Work

In artificial intelligent community, a lot of effort has been devoted to develop reasoning mechanisms dealing with uncertainty [12, 14, 13]. Here, the focus was on determining how subtle uncertainties affect the reasoning mechanisms. Differently, the autonomous

agent community is mainly focusing on how to implement agent platforms (e.g., Jack, Jason and Jadex) with reasoning mechanisms to deal with existing event and situation [30, 31, 20].

Helpern [12] proposes a framework to reasons about uncertainty from a single agent viewpoint and multi-agent environments. An environment (i.e., entities outside the agent) is treated as black boxes, with no structure. Therefore, the likelihood of events is viewed as a collection of random variables. In the multi-agent framework, the environment can be structured on the basis of the interactions among agents. The framework also models uncertainty values in terms of functions of time.

Markov Decision Process (MDPs) [13] is a mathematical framework to model a decision-making process where outcomes are partly random and partly controlled by agents. MDPs address optimization problems. They are represented as tuples $\langle S, A, P, R \rangle$, where S is a set of possible states, A is a set of actions that caused state transitions, P is a set of probabilities of action occurrences, and R is a set of rewards that are gained if an action is executed. The framework uses value iteration algorithms to determine the set of actions that minimize and maximize the reward. If risks are encoded as rewards, the agent can use such algorithms to identify the least risk strategy by searching the solution with minimal reward. Moreover, Samari and Parson [32] investigated on the relation between MDPs and BDI architectures. Their results open the way for using MDPs as reasoning mechanisms in the BDI architecture.

Another proposal is the Abstract Agent Programming Language[3] (3APL), a cognitive agent programming language that employs logical representation and reasoning features based on the BDI model. In this setting, the beliefs are assumed to be certain, either true or false. Kwisthout and Dastani [14] propose an extension of 3APL that allows an agent to reason about uncertain beliefs. The authors recognized that 3APL is not sufficient to reason about uncertain beliefs that result essential for dealing with many real cases. This extension uses Dempster-Shafer theory [23] to model uncertainty in agent's beliefs. Those reasoning frameworks are really sufficient for reasoning about uncertainty, even sometime it is too complex, but those frameworks does not specify how an agent must react to deal with the effects of uncertainty.

In the autonomous agent community, the proposals, such as [31, 20], present implementations of the UAV agent (called Wingman) in the Jack platform. The Wingman is provided with basic plans (e.g., flying quietly and at low altitude) in order to achieve its goals (e.g., taking picture of enemy installations). The Wingman is also provided with additional plans (e.g., flying as fast as possible) to deal with malicious events such as being detected by enemy. When a malicious event occurs, the Wingman reasons about its plans and changes the adopted basic plan with an additional plan to guarantee the achievement its goals. However, the Wingman does not try to anticipate the occurrence of malicious events, but it just reacts when they happen. Similar works have been done in [18, 33]. On the contrary, our approach allows the UAV agent to anticipate malicious events by employing treatments when the risk is unacceptable. Another implementation of reasoning in an autonomous agent has been proposed by Braubach et al. [30]. They propose an implementation of the Cleaner World agent in Jadex. Such an agent is designed to pick all wastes and throw them in trash bins. In this implementation, the

[3] http://www.cs.uu.nl/3apl/

agent moves to a location once its sensors detect the existence of wastes. However, it would be more efficient if agent movements are not only driven by the appearance of wastes, but they are anticipated by reasoning about the likelihood of having new wastes in a certain location.

Finally, this paper propose a middle way between those two approaches to reason about uncertainty and react accordingly. Our work starts from using the existing reasoning mechanism in the agent platforms (i.e., Jadex), and extends it with Goal-Risk reasoning mechanism [22]. This approach is taken because we intend to have the reasoning mechanism which is implementable in an agent platform, especially the ones that are based on BDI. Moreover, the reasoning looks so trivial comparing with the ones that are proposed by AI community. the GR framework is not meant to assess the uncertainty precisely, but it is used for supporting an agent in constructing a strategy which is not risky. Thus, the GR framework can use a simple metric to calculate uncertainty.

6 Conclusions and Remarks

In this paper, we have presented a Jadex implementation of the Tropos Goal-Risk framework. The GR framework and Jadex differ in concepts and reasoning features and their integration was not straightforward. In particular, we have shown that there are limitations in Jadex to adopt all concepts supported by the GR framework, but we have also shown that this mapping does not limit the expressiveness of the intended goal deliberation process.

The initial GR reasoning mechanisms allow agents to perform cost-benefit analysis on a strategy to be adopted, but require an exhaustive knowledge before the agent starts pursuing its goals. On the other hand, the Jadex approach results faster in choosing a strategy for a given goal and more adaptive to the current conditions of the agent. We have taken advances of combining GR and Jadex reasoning features by implementing complete plans and facilities for reasoning about them in the belief of the agent. The intuition is to use complete plans as a baseline for the Jadex reasoning. Then, during the pursuing of its goals, the agent evaluates the baseline by reasoning about risks in the current situation.

We are currently extending the risk reasoning to cope with multi-agent environments. In this setting, an agent should be able to reason about risks when it depends on other agents for the fulfillment of its goals. Yet, treatments may be introduced when risks are unacceptable or when the agent has no alternatives to achieve its goal besides depending on other agents.

Acknowledgments

We thank Jan Sudeikat for many useful discussions on Jadex platform . This work has been partially funded by EU-SENSORIA and EU-SERENITY projects, by FIRB-TOCAI project, and by PAT-MOSTRO project.

References

1. Lauber, J., Steger, C., Weiss, R.: Autonomous Agents for Online Diagnosis of a Safety-critical System Based on Probabilistic Causal Reasoning. In: Proceedings of the The Fourth International Symposium on Autonomous Decentralized Systems (ISADS '99), Washington, DC, USA, IEEE Computer Society (1999) 213–219
2. Kumar, S., Cohen, P.R.: Towards a Fault-Tolerant Multi-Agent System Architecture. In: Proceedings of the Fourth International Conference on Autonomous Agents (AGENTS '00), New York, NY, USA, ACM Press (2000) 459–466
3. McCarthy, J.: Ascribing Mental Qualities to Machines. Technical Report Memo 326, Stanford AI Lab, Stanford (1979)
4. Rao, A.S., Georgeff, M.P.: BDI Agents: From Theory to Practice. In: Proceedings of 1st International Conference on Multi-Agent Systems (ICMAS '95). (1995) 312–319
5. Shoham, Y.: Agent-Oriented Programming. Artificial Intelligence **60**(1) (1993) 51–92
6. Bratman, M.: Intention, Plans, and Practical Reason. Harvard University Press (1987)
7. Rao, A.S., Georgeff, M.P.: Modeling Rational Agents within a BDI-Architecture. In: Proceedings of 2nd International Conference on Principles of Knowledge Representation and Reasoning (KR'91), Morgan Kaufmann publishers Inc. (1991) 473–484
8. Howden, N., Ronnquist, R., Hodgson, A., Lucas, A.: JACK Intelligent Agents-Summary of an Agent Infrastructure. In: Proceedings of the 5th International Conference on Autonomous Agents (AGENTS '01), ACM Press (2001)
9. Pokahr, A., Braubach, L., Lamersdorf, W.: Jadex: A BDI Reasoning Engine. In: Multi-Agent Programming: Languages, Platforms and Applications. Springer Science+Business Media Inc., USA (September 2005) 149–174
10. Bordini, R.H., Hübner, J.F.: BDI Agent Programming in AgentSpeak Using Jason. In: Proceedings of 6th International Workshop on Computational Logic in Multi-Agent Systems (CLIMA VI). Volume 3900 of LNCS., Springer (2005) 143–164
11. COSO: Enterprise Risk Management - Integrated Framework. Committee of Sponsoring Organizations of the Treadway Commission. (September 2004)
12. Halpern, J.Y.: Reasoning About Uncertainty. The MIT Press (2003)
13. White, D.J.: Markov Decision Processes. John Wiley & Sons (1993)
14. Kwisthout, J., Dastani, M.: Modelling Uncertainty in Agent Programming. In: Proceedings of Third International Workshop on Declarative Agent Languages and Technologies III (DALT '05). Volume 3904 of LNCS., Springer (2005) 17–32
15. Asnar, Y., Giorgini, P., Mylopoulos, J.: Risk Modelling and Reasoning in Goal Models. Technical Report DIT-06-008, DIT - University of Trento (February 2006)
16. Giorgini, P., Mylopoulos, J., Sebastiani, R.: Goal-Oriented Requirements Analysis and Reasoning in the Tropos Methodology. Engineering Applications of Artificial Intelligence **18**(2) (March 2005) 159–171
17. Feather, M.S., Cornford, S.L., Hicks, K.A., Johnson, K.R.: Applications of tool support for risk-informed requirements reasoning. Computer Systems Science & Engineering **20**(1) (January 2005) 5–17
18. Dufrene Jr., W.R.: Approach for Autonomous Control of Unmanned Aerial Vehicle Using Intelligent Agents for Knowledge Creation. In: Proceedings of The 23rd Conference on Digital Avionics Systems Conference (DASC '04). Volume 2. (2004) 1–9
19. Karim, S., Heinze, C.: Experiences with the Design and Implementation of an Agent-Based Autonomous UAV Controller. In: Proceedings of 4th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS '05), ACM Press (2005) 19–26
20. Wallis, P., Ronnquist, R., Jarvis, D., Lucas, A.: The Automated Wingman - Using JACK Intelligent Agents for Unmanned Autonomous Vehicles. In: Proceedings of IEEE Aerospace Conference. Volume 5. (2002) 2615–2622

21. Bresciani, P., Perini, A., Giorgini, P., Giunchiglia, F., Mylopoulos, J.: Tropos: An Agent-Oriented Software Development Methodology. Journal of Autonomous Agents and Multi-Agent Systems **8**(3) (2004) 203–236
22. Asnar, Y., Giorgini, P.: Modelling Risk and Identifying Countermeasures in Organizations. In: Proceedings of 1st International Workshop on Critical Information Infrastructures Security (CRITIS '06). Volume 4347 of LNCS., Springer (2006) 55–66
23. Shafer, G.: A Mathematical Theory of Evidence. Princeton University Press, Princeton, NJ (1976)
24. Giorgini, P., Mylopoulos, J., Nicchiarelli, E., Sebastiani, R.: Formal Reasoning Techniques for Goal Models. Journal of Data Semantics (October 2003)
25. Bedford, T., Cooke, R.: Probabilistic Risk Analysis: Foundations and Methods. Cambridge University Press (2001)
26. Stamatelatos, M., Vesely, W., Dugan, J., Fragola, J., Minarick, J., Railsback, J.: Fault Tree Handbook with Aerospace Applications. NASA (2002)
27. Cook, S.A., Mitchell, D.G.: Finding Hard Instances of the Satisfiability Problem: A Survey. In: Satisfiability Problem: Theory and Applications. Volume 35. American Mathematical Society (1997) 1–17
28. Sebastiani, R., Giorgini, P., Mylopoulos, J.: Simple and Minimum-Cost Satisfiability for Goal Models. In: Proceedings of the 16th Conference On Advanced Information Systems Engineering (CAiSE '04). Volume 3084 of LNCS., Springer (2004) 20–33
29. Georgeff, M., Lansky, A.: Reactive Reasoning and Planning. In: Proceedings of the Sixth National Conference on Artificial Intelligence (AAAI '87), Seattle, WA, Morgan Kaufmann (1987) 677–682
30. Braubach, L., Pokahr, A., Lamersdorf, W., Moldt, D.: Goal Representation for BDI Agent Systems. In: Proceedings 2nd International Workshop on Programming Multiagent Systems: Languages and Tools (ProMAS '04). Volume 3346 of LNAI., Springer (2004) 9–20
31. Karim, S., Heinze, C., Dunn, S.: Agent-Based Mission Management for a UAV. In: Proceedings of the 2004 of Intelligent Sensors, Sensor Networks and Information Processing Conference (ISSNIP '04), IEEE Press (2004) 481–486
32. Simari, G.I., Parsons, S.: On the Relationship between MDPs and the BDI Architecture. In: Proceedings of 5th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS '06), ACM Press (2006) 1041–1048
33. Vidolov, B., De Miras, J., Bonnet, S.: AURYON - A Mechatronic UAV Project Focus on Control Experimentations. In: Proceedings of the International Conference on Computational Intelligence for Modelling, Control and Automation and International Conference on Intelligent Agents, Web Technologies and Internet Commerce (CIMCA-IAWTIC '06). Volume 1., Washington, DC, USA, IEEE Computer Society (2005) 1072–1078