

# IBM Scenario Planning Advisor: Plan recognition as AI planning in practice

Shirin Sohrabi<sup>a</sup>, Michael Katz<sup>a,\*</sup>, Oktie Hassanzadeh<sup>a</sup>, Octavian Udrea<sup>a</sup>, Mark D. Feblowitz<sup>a</sup> and Anton Riabov<sup>b</sup>

<sup>a</sup> IBM Research, 1101 Kitchaawan Rd, Yorktown Heights, NY 10598, United States

E-mail: [michael.katz1@ibm.com](mailto:michael.katz1@ibm.com)

<sup>b</sup> Logitech Inc., 7700 Gateway Blvd, Newark, CA 94560, United States

**Abstract.** We present the IBM Research Scenario Planning Advisor (SPA), a decision support system that allows users to generate diverse alternate scenarios of the future and enhance their ability to imagine the different possible outcomes, including unlikely but potentially impactful futures. Our system, takes as input the relevant information from news and social media, representing key risk drivers, as well as the domain knowledge and generates scenarios that explain the key risk drivers and describe the alternative futures. To this end, we provide a characterization of the problem, knowledge engineering methodology, and transformation to AI planning. Furthermore, we describe the computation of the scenarios, lessons learned, and the feedback received from the pilot deployment of the SPA system in IBM.

Keywords: AI planning, scenario planning, risk management, plan recognition

## 1. Introduction and motivation

Scenario planning is a widely accepted technique by which organizations develop their long-term plans [14]. Scenario planning for risk management puts an added emphasis on identifying the extreme yet possible risks and opportunities that are not usually considered in daily operations. Scenario planning involves analyzing the relationship between forces — such as social, technical, economic, environmental, and political trends — in order to explain the current situation and to provide insights about the future. A major benefit to scenario planning is that it helps businesses or policy-makers to learn about possible alternative futures and to anticipate them [9]. We use AI planning, informed by expert domain knowledge, because some scenarios have never yet occurred and thus cannot be projected by probabilistic means. And we generate many different scenarios, exploring a variety of possible futures; because we want to be prepared for both expected and surprising futures.

Risk management is a set of principles that focus on the outcome for risk-taking [20]. A variety of (manual) methods and standards for risk management have been developed [2]. Our approach in addressing sce-

nario planning for risk management is different from previous work in that we reason about emerging risks based on observations from the news and social media trends, and produce scenarios that both describe the current situation and project the future possible effects of these observations. Our objective is to compute multiple alternate scenarios, informing the decision-makers of the breadth of possibilities that may need consideration. This is different from a narrow focus on predicting the most likely outcome. Furthermore, each scenario we produce highlights the potential *leading indicators*, the set of facts that are likely to lead to a scenario, the *scenario and emerging risk*, the combined set of consequences or effects in that scenario, in addition to the *business implications*, a subset of potential effects of that scenario that the users care about and are relevant to the enterprise. For example, prior to the Brexit referendum in 2016, an international company operating in the United Kingdom could consider alternative future scenarios for changes in trade and employment treaties assuming the majority voted to leave the European Union, identifying the implications for the company's finances and its ability to hire, enabling the company to act immediately to minimize the negative implications.

The main functions of Scenario Planning Advisor (SPA) are: (1) discovering active risk drivers by ag-

---

\* Corresponding author. E-mail: [michael.katz1@ibm.com](mailto:michael.katz1@ibm.com).

gregating relevant news from the Web and social media, and generating lists of candidate observations corresponding to the detected risk drivers;<sup>1</sup> (2) generating multiple alternative scenarios based on the user-selected observations, the domain knowledge about the driver relations, and the potential likelihoods and impact of their cascading effects.<sup>2</sup> The main idea of the approach in SPA is to characterize the scenario planning problem for enterprise risk management as a plan recognition problem [19] and use AI planning to address the plan recognition problem [11,12,15,16]. To this end, we transform the domain knowledge as captured by Mind Maps, a graphical representation that captures concepts and their relations, into an AI planning task. Furthermore, we generate multiple high-quality plans [7,13,17] and cluster them into scenarios. These scenarios are then used to start a risk conversation between the analysts and decision makers. To the best of our knowledge, we are the first to apply AI planning in addressing scenario planning for enterprise risk management. We believe that AI planning provides a very natural formulation for the efficient exploration of possible outcomes required for scenario planning.

## 2. Preliminaries

In this section, we briefly review the necessary background on AI planning and Plan Recognition. We consider planning tasks  $\Pi = \langle F, A, I, G, \text{COST} \rangle$  in the STRIPS formalism extended with action costs. In such a task,  $F$  is a set of Boolean *fluents*. Each subset  $s \subseteq F$  is called a *state*, and  $S(\Pi) = 2^F$  is the *state space* of  $\Pi$ . The state  $I$  is the *initial state* of  $\Pi$ . The goal  $G \subseteq F$  is a set of fluents, where a state  $s$  is a *goal state* if  $G \subseteq s$ .  $A$  is a finite set of *actions*, each having an associated set of *preconditions*  $\text{pre}(a) \subseteq F$ , *add effects*  $\text{add}(a) \subseteq F$  and *delete effects*  $\text{del}(a) \subseteq F$ , and  $\text{COST} : A \rightarrow \mathbb{R}^{0+}$  being a non-negative *action cost* function.

The semantics of STRIPS planning is as follows. An action  $a$  is *applicable* in the state  $s$  if  $\text{pre}(a) \subseteq s$ . Applying  $a$  in  $s$  results in the state  $s[a] := (s \setminus \text{del}(a)) \cup \text{add}(a)$ . A sequence of actions  $\pi = \langle a_1, \dots, a_k \rangle$  is *applicable* in  $s$  if there exists a sequence of states  $\langle s_0, \dots, s_k \rangle$  such that  $s_0 = s$ , action  $a_i$  is applicable in state  $s_{i-1}$ , and  $s_i = s_{i-1}[a_i]$ . If it exists, such a path is uniquely defined, and its end state is denoted by  $s[\pi]$ .

<sup>1</sup>Demo video is here: <https://youtu.be/6BRXwSkRC04>.

<sup>2</sup>Demo video is here: <https://youtu.be/N0AsRbdL7WM>.

An applicable action sequence is a *plan* for  $s$  if  $s[\pi]$  is a goal state. Its *cost* is the cumulative cost of actions in the sequence:  $\text{COST}(\pi) = \sum_{i=1}^k \text{COST}(a_i)$ . A plan for  $s$  with minimal cost is called *optimal*. The objective of (optimal) planning is to find an (optimal) plan for  $I$ .

A Plan Recognition (PR) problem over a domain theory is a tuple  $R = \langle \Pi, O, \mathcal{G}, \text{PROB} \rangle$ , where  $\Pi = \langle F, A, I, G, \text{COST} \rangle$  is a planning task,  $O = \{o_1, \dots, o_m\}$ ,  $o_i \in F$ ,  $i \in [1, m]$  is a set of observations,  $\mathcal{G} \subseteq S(\Pi)$  is the set of possible goals, and  $\text{PROB}$  is a probability distribution over the goals  $\mathcal{G}$ . Note, this definition includes a minor modification from previous work [11,15] as it includes the planning task  $\Pi$  as the input to the plan recognition problem.

The solution to the PR problem is a pair of probability distributions. The first is the probability of plans given the observations,  $P(\pi|O)$ , where each  $\pi$  is a plan for  $\Pi$  that traverse through at least one goal  $s \in \mathcal{G}$  for the plan recognition problem, and satisfies the observation sequence,  $O$ . The second is the probability of the plan recognition problem goals, given the observations,  $P(s|O)$ , where  $s \in \mathcal{G}$ . Note, an observation is said to be satisfied by an action sequence if it is either explained or discarded following the work of Sohrabi et al. [15]. This allows for some observations to be left unexplained, in particular if they are out of context with respect to the rest of the observations.

In previous work, AI planning is used to approximate these probabilities. In [15],  $P(\pi|O)$  is approximated by considering the cost of the original actions, as well as the number of unexplainable observations. Posterior probabilities of goals given observations,  $P(s|O)$ , are then computed by a summation over  $P(\pi|O)$  for all plans that achieve  $s$  and satisfy  $O$ . Posterior probabilities of goals given observations,  $P(s|O)$ , can also be computed by considering the cost difference of plans, or  $\Delta$ , that achieve the goal  $s$  and the observations  $O$  and achieve the goal  $s$ , but not the observations  $O$ , as in [12].

## 3. Scenario Planning Advisor (SPA)

The architecture for our system, Scenario Planning Advisor (SPA), is shown in Fig. 1. There are three major components. The planning engine, shown under the *Scenario Generation and Presentation* component, takes as input the output of the other two components: the *News Aggregation* and the *Domain Knowledge*. The *News Aggregation* component deals with analyzing the raw data coming from the news and social

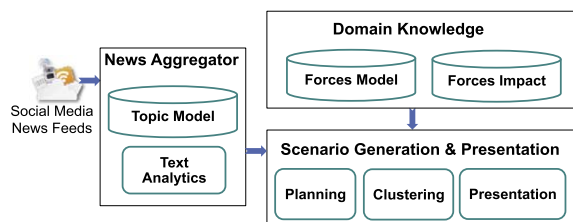


Fig. 1. The SPA system architecture.

media feeds. The Topic Model, provided by the domain expert, includes the list of important people, organizations, and keywords. The result of the *News Aggregation* component is a set of relevant *key risk drivers* or observations, a subset of which can be selected by the business user and is fed into the *Scenario Generation and Presentation* component.

The *Domain Knowledge* component captures the necessary domain knowledge in two forms, Forces Model and Forces Impact. The Forces Model is a description of the causes and consequences for a certain force, such as social, technical, economic, environmental, and political trends, and is provided by a domain expert who have little or no AI planning background. Forces Model are captured by a set of Mind Maps ([https://en.wikipedia.org/wiki/Mind\\_map](https://en.wikipedia.org/wiki/Mind_map)), a graphical representation that encodes concepts and relations. The Forces Impact, describes potential likelihoods and impact of a cause or a consequence. Forces Impact also describes the level of importance of a main force. Business implications is a set of predefined concepts (e.g., the concepts that mention the name of the company). The *Scenario Generation* component takes the domain knowledge and the key risk drivers and automatically generates a planning problem whose solutions, when clustered in the post-processing step induce a set of alternative scenarios.

Hence, we define the scenario planning problem for enterprise risk management as a tuple  $SP = \langle Forces\ Model, Forces\ Impact, Key\ Risk\ Drivers \rangle$ . Key Risk Drivers are a subset of forces describing the current situation as suggested by the News Aggregation component. Solutions to the  $SP$  problem are sets of alternative scenarios that consider the key risk drivers and describe a range of possible futures considering the likelihood, impact and importance values based on the Forces Model and Forces Impact. Each such scenario corresponds to a multi-path through the *Forces Model*, traversing *Key Risk Drivers*. The  $SP$  problem is computationally hard, as can be shown by, e.g., a polynomial reduction from the Hamiltonian Path problem. Thus, in what follows, we exploit planning to solve it.

## 4. News aggregation

SPA needs to be constantly aware of an evolving world, filtered and aggregated based on users' interest profiles. For that, SPA continuously monitors multiple real-world sources (e.g., news channels from RSS and Atom feeds, social media posts) in multiple languages. To this end, several text analytics are implemented to find the information relevant for a particular domain in the vast amount of information available to crawl. To further refine and filter the information, SPA uses the structured semantic knowledge available in Wikidata using Wikidata Query Service to find important people, organization, and relevant sources [18] and stores these in the Topic Model. Analysts then review the generated results and select key risk drivers that are the most relevant and important for them. Note that SPA can deal with unreliable observations (i.e., noisy, inconsistent, missing observations) as it exploits previous work on plan recognition as planning [15]. Hence, SPA is not forced to explain all the selected key risk drivers but is encouraged to explain as many observations as possible.

## 5. Domain knowledge

The Mind Maps (i.e., the Forces Model) and the customization components (i.e., Forces Impact) encode knowledge about risk drivers and business implications elicited from the domain experts and the local country experts correspondingly. While the reasoning engine in SPA supports a rich representation of risk drivers as actions in Planning Domain Description Language (PDDL) [8], the knowledge representation used by domain experts is drastically simplified, to prevent conflicts and reduce overheads in knowledge elicitation and maintenance. The domain experts use Mind Maps created in FreeMind ([freemind.sourceforge.net/wiki/](http://freemind.sourceforge.net/wiki/)), a graphical tool that encodes concepts and relations, to capture directed graphs of risk drivers and business implications, with edges having hidden semantics of pairwise cause and effect.

### 5.1. Forces model as structured Mind Map

We represent the Forces Model as a set of Mind Maps. Two example Mind Maps are shown in Fig. 2. The main forces in these Mind Maps are the “currency depreciation against US dollar” and the “decrease in price of commodity”. The forces with an edge going

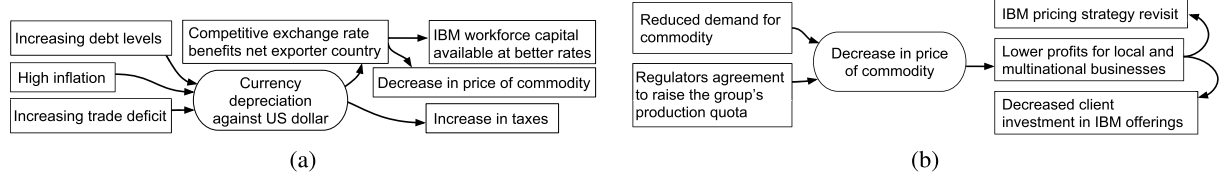


Fig. 2. Part of the Mind Maps for (a) currency depreciation against US dollar (b) decrease in price of commodity. Edges indicate the direction of the causes and effects. Concepts that have the name of the company in them (e.g., IBM) are the business implications.

towards the main force, are the possible causes, and the forces with an outgoing edge from another force, are the possible consequences. The causes and effects can appear in chains, and cascade to other causes, and effects, with a leaf node of either a business implication, or another force, with its own separate Mind Map that describes it. For example, the leaf node “IBM workforce capital available at better rates” is an example of a business implication, and the leaf node “Decrease in price of commodity” is itself a main force described in the Mind Map in Fig. 2(b).

Next, we formally define Mind Maps. Let  $B$  and  $C$  be two disjoint sets, where  $B$  is a set of symbols of type business implications and  $C$  is a set of symbols of type force. We now define a set of structured Mind Maps.

**Definition 1.** A set of structured Mind Maps  $\mathcal{M}$  is a set of tuples  $M = \langle \Gamma, \sigma, \Theta \rangle$ , where  $\Gamma$  is a causal structure for  $M$ ,  $\sigma \in C$  is the main force, and  $\Theta$  is a consequence structure for  $M$ . A causal structure  $\Gamma$  is defined as a set of causal sequences such that each sequence takes one of the following forms:

- $[c_1, \dots, c_m, \sigma]$ , where  $c_i \in C$ ,  $1 \leq i \leq m$ , or
- $[c_1, \dots, c_i, c_{i+1}, \dots, c_m, \sigma]$ , where  $[c_1, \dots, c_i] \in \Gamma'$ , for some structured Mind Map  $M' = \langle \Gamma', c_i, \Theta' \rangle$ ,  $M' \in \mathcal{M}$ , and  $c_{i+1}, \dots, c_m \in C$ , for some  $1 \leq i \leq m$

Further, a consequence structure  $\Theta$  is defined as a set of consequence sequences such that each consequence sequence takes one of the following forms:

- $[\sigma, c_1, \dots, c_{n-1}, c_n]$ , where  $c_i \in C$ ,  $1 \leq i < n$ ,  $c_n \in B$
- $[\sigma, c_1, \dots, c_i, c_{i+1}, \dots, c_n]$ , where  $c_1, \dots, c_i \in C$ ,  $[c_{i+1}, \dots, c_n] \in \Theta''$  for some structured Mind Map  $M'' = \langle \Gamma'', c_{i+1}, \Theta'' \rangle$ ,  $M'' \in \mathcal{M}$ , for some  $0 \leq i < n$ .

Going back to our example in Fig. 2, causal structure defines the structure of the nodes and edges to the left of the main force, i.e., all nodes with an arrow to the main force (e.g., “High inflation”), and the consequence structure defines the structure of the nodes and

edges to the right of a main force, i.e., nodes with an arrow from the main force and all subsequent nodes, (e.g., “IBM pricing strategy revisit”). Note that this definition is recursive.

We can now define the notion of a path in Mind Maps.

**Definition 2.** Given a set of structured Mind Maps,  $\mathcal{M}$ , a valid path  $\varphi$  is a sequence of symbols  $[c_1, \dots, c_{i-1}, c_i, c_{i+1}, \dots, c_{n-1}, c_n]$ ,  $c_1, \dots, c_{n-1} \in C$ ,  $c_n \in B$ , such that for each  $1 < i < n$ , there exists a Mind Map  $M = \langle \Gamma, c_i, \Theta \rangle$ ,  $M \in \mathcal{M}$ , where  $[c_1, \dots, c_{i-1}] \in \Gamma$ , and  $[c_{i+1}, \dots, c_{n-1}, c_n] \in \Theta$ .

Informally, a valid path through the set of Mind Maps starts from the causal structure, goes through at least one main force, and ends in a business implication symbol. Thus a valid path consists of at least three nodes. In Fig. 2, the path “Reduced demand for commodity”, “Decrease in price of commodity”, “Lower profits for local and multinational businesses”, “IBM pricing strategy revisit” is a valid path. So is the path “High inflation”, “Currency depreciation against US dollar”, “Competitive exchange rate benefits net exporter country”, “Decrease in price of commodity”, “Lower profits for local and multinational businesses”, “Decreased client investment in IBM offerings”. Note, Mind Maps can be connected through both cause and consequence sequences; that is, many main forces can appear on a valid path. Also, many valid paths exist for a given set of structured Mind Maps. The additional information provided by the Forces Impact allows us to rank these paths.

## 5.2. Forces impact via questionnaire

Additional information on the Mind Maps is encoded through the Forces Impact. The Forces Impact, describes potential likelihoods and impact of a cause (i.e., nodes with an edge going into the main force) or a consequence (e.g., nodes with an edge going from the main force and all other cascading nodes). Forces Im-

pact also describes the level of importance of a main force.

One way to capture this information, and the approach we take, is to ask the domain experts a series of automatically generated questions based on the Mind Maps. For example, the system will ask the following question in order to understand which of the causes are more likely: “How likely are any of the following to lead to currency depreciation against US dollar.” The system will also ask the following question in order to understand which consequences are more likely and would have a higher impact: “Assuming currency depreciation against US dollar occurs, please evaluate the likelihood and impact of the following effects.” In addition, the system will ask the domain expert to specify the relative importance of the main forces in their particular situation (i.e., company). Importance, impact, and likelihood can take one of the values: low, medium, or high. This can be easily extended to any finite number of values.

More formally, given a set of Mind Maps  $\mathcal{M}$ , let  $\Sigma$  be the set of all possible pairs of symbols, where for each pair  $r \in \Sigma$ , there exists a Mind Map  $M = \langle \Gamma, \sigma, \Theta \rangle$ , such that  $r$  appears in  $\Gamma$  or  $\Theta$ . We denote  $r^{\text{impact}}$  and  $r^{\text{likelihood}}$  to denote the impact and likelihood of that pair (i.e., edge in a Mind Map). Also, we denote,  $M^{\text{importance}}$ , to be the level of importance for a given structured Mind Map  $M \in \mathcal{M}$ .

Given this additional information on the Mind Maps, we can define a ranking among valid paths. Informally, valid paths that go through Mind Maps with high importance value, causes and consequences with high impact and likelihood have a higher quality. In the next section, we describe how these values can be encoded with action costs such that a high-quality valid path would map to a low-cost plan.

## 6. Transformation to planning

Since the scenario planning problem, as mentioned in Section 3, is NP-hard, in this section we describe our solution using planning. Given a scenario planning problem, we define its corresponding plan recognition problem, which allows us to apply the previous work on plan-recognition-as-planning to generate many plans. In addition, we will describe our method of translating the domain knowledge into the planning task.

**Definition 3.** Given a scenario planning problem,  $SP$ , as described in Section 3, a corresponding plan recognition problem is defined as a tuple  $SPPR = \langle \Pi, O, \mathcal{G}, \text{PROB} \rangle$ , for a planning task  $\Pi = \langle F, A, I, G, \text{COST} \rangle$  described by the *Forces Model* and Forces Impact, with the set of observations that consists of the selected Key Risk Drivers, the set of possible goals  $\mathcal{G}$  that consists of the business implications as specified in the Forces Model, and  $\text{PROB}$  is the uniform probability distribution over the set of possible goals.

Given the corresponding plan recognition problem  $SPPR$ , we follow the plan-recognition-as-planning approach [15] that approximates the posterior probabilities of goals and plans by computing a set of plans. However, instead of computing the posterior probabilities of goals and plans, which is not the objective of the scenario planning problem, we group the set of computed plans and present the grouping as scenarios to the users.

**Definition 4.** Given a scenario planning problem,  $SP$ , and its corresponding plan recognition problem,  $SPPR$ , as defined above, solutions to  $SPPR$  problem are sets of scenarios, where each scenario is a collection of plans such that each plan  $\pi$ : (i) traverses a state that meets at least one of the possible goals (i.e.,  $\exists G' \in \mathcal{G}$ , where  $G' \subseteq s$ ) and (ii) satisfies the set of observations (i.e., observations are either explained or discarded).

Informally, scenarios group plans by a certain similarity criteria, e.g., sets of facts that are true in the end state. We further elaborate on that in Section 7. Note that a set of scenarios or a solution to the  $SPPR$  problem also formally defines a solution to the scenario planning problem,  $SP$ , as described in Section 3.

Next, we will describe how to translate the set of Mind Maps  $\mathcal{M}$  together with their importance level, impact and likelihoods into a planning task. We will also show that a valid path maps directly to a plan for the planning task. Note that the (*is-true*)predicate ensures that only one indicator action is executed for each valid path.

**Definition 5.** Given a set of Mind Maps  $\mathcal{M}$ , their importance level  $M^{\text{importance}}$ ,  $M \in \mathcal{M}$ , set of all possible pairs of symbols  $\Sigma$ , and their impact and likelihood levels,  $r^{\text{impact}}$ ,  $r^{\text{likelihood}}$ ,  $r \in \Sigma$ , we define a planning task  $\Pi = \langle F, A, I, G, \text{COST} \rangle$  as follows:

- $F$  is a set of fluents that appear in  $A$ :
  - \* (*is-true*),
  - \* (*achieved*),

- \* (*bis*  $c$ ) for all  $c \in B$ ,
  - \* (*at*  $c$ ) for all  $c \in C$ ,
  - \* (*low*  $c_1$   $c_2$ ), (*med*  $c_1$   $c_2$ ), (*high*  $c_1$   $c_2$ ) for all  $c_1, c_2 \in C$ , corresponding to the combined values of  $r^{\text{impact}}$ ,  $r^{\text{likelihood}}$  for the pair  $r = (c_1, c_2)$ , and
  - \* (*f-low*  $c$ ), (*f-med*  $c$ ), (*f-high*  $c$ ), for all  $c \in C$ , where  $c$  is a main force for one of the Mind Maps  $M \in \mathcal{M}$ , corresponding to  $M^{\text{importance}}$ .
- $A$  is the union of the following action sets:
    - \*  $A_{\text{next-low}}$ , for each pair  $(c_1, c_2) \in \Sigma$ , with precondition (*low*  $c_1$   $c_2$ ) and (*at*  $c_1$ ), add effects (*at*  $c_2$ ), delete effects (*at*  $c_1$ ), and cost corresponding to the combined values of  $r^{\text{impact}}$ ,  $r^{\text{likelihood}}$ ,
    - \*  $A_{\text{next-med}}$ ,  $A_{\text{next-high}}$ , similar to  $A_{\text{next-low}}$ , where low is replaced with med and high respectively,
    - \*  $A_{\text{nextbis}}$ , for each pair  $(c_1, c_2) \in \Sigma$ , where  $c_2 \in B$ , with precondition (*at*  $c_1$ ), add effects (*bis*  $c_2$ ), delete effects (*at*  $c_1$ ), and a cost corresponding to the combined values of  $r^{\text{impact}}$ ,  $r^{\text{likelihood}}$ ,
    - \*  $A_{\text{indicator-low}}$ , for each causal sequence  $[c_1, \dots, c_n]$  as defined in  $M \in \mathcal{M}$ , with precondition (*f-low*  $c_n$ ) and (*is-true*), add effects (*at*  $c_1$ ), delete effects (*is-true*), and a cost corresponding to  $M^{\text{importance}}$ ,
    - \*  $A_{\text{indicator-med}}$ ,  $A_{\text{indicator-high}}$  where low is replaced with med and high respectively, and
    - \*  $A_{\text{achieve-goal}}$  for each  $c \in B$ , with precondition (*bis*  $c$ ), add effect (*achieved*), no delete effect, and zero cost.
  - $I = \{(is-true), (low\ c_1\ c_2), (med\ c_1\ c_2), (high\ c_1\ c_2), (f-low\ c), (f-med\ c), (f-high\ c)\}$ , as defined by  $F$ .
  - $G = \{(achieved)\}$ .

**Theorem 1** (Soundness/Correctness). *Given a set of Mind Maps  $\mathcal{M}$  and the corresponding planning task  $\Pi$  as defined above, if  $\varphi$  is a valid path for  $\mathcal{M}$ , then we can construct a sequence of actions  $\pi$ , such that  $\pi$  is a plan for the planning task  $\Pi$ . On the other hand, if  $\pi$  is a plan for the planning task  $\Pi$ , then there exists a valid path  $\varphi$  for  $\mathcal{M}$ , where  $\varphi$  can be constructed from  $\pi$ . Furthermore, a valid path  $\varphi_1$  has a higher quality than a valid path  $\varphi_2$  if and only if  $\text{COST}(\pi_1) < \text{COST}(\pi_2)$  for the corresponding plans  $\pi_1$  and  $\pi_2$ .*

**Proof.** ( $\Rightarrow$ ) Given a valid path  $\varphi = \langle c_1, \dots, c_{i-1}, c_i, c_{i+1}, \dots, c_{n-1}, c_n \rangle$ , we construct a plan  $\pi$  for the planning task  $\Pi$  as follows. The first action is an *indicator* action for  $\langle c_1, \dots, c_{i-1}, c_i \rangle$ . Then, a sequence of *next* actions, one for each pair of symbols in the path, followed by a *nextbis* action for the pair  $(c_{n-1}, c_n)$ . Finally, an *achieve-goal* action for the business implication  $c_n \in B$  achieves the goal of the planning task.

( $\Leftarrow$ ) Given a plan  $\pi$  for the planning task  $\Pi$ , we construct a valid path for  $\mathcal{M}$ , considering the arguments of the actions. We also must make sure that the cost of the actions corresponds to the importance, impact and likelihood values.  $\square$

The translation method described above could have different implementations. In particular, to determine the costs associated with the combined values for likelihood and impact, different methods can be used. For example, to combine likelihood and impact, one can consider a high value, if both the likelihood or impact are high, a medium value if either values are high, or both are medium, and a low value otherwise. The low/medium/high can also map to any numbers in the cost of the action. However, as long as their relative difference adheres to the three levels, where low maps to a higher cost and vice versa, the theorem holds.

We can directly represent the transformed planning task in a “lifted” planning language such as PDDL [8] where we would define one general and “lifted” set of actions in the domain file, defining problem files based on the given Mind Maps. As a generic grounding algorithm may take a substantial amount of time, we also experiment with creating directly a (partially) grounded planning task. To obtain such (partially) grounded planning task, we fully ground the “next” and “indicator” actions, alleviating the need for the static predicates (*low*  $c_1$   $c_2$ ), (*med*  $c_1$   $c_2$ ), and (*high*  $c_1$   $c_2$ ), as well as (*f-low*  $c$ ), (*f-med*  $c$ ), and (*f-high*  $c$ ). All other actions remain lifted. We evaluate the performance of both methods in the experimental evaluation section.

## 7. Plan and scenarios computation

In the previous section we discussed a sound and complete translation of Forces Models and Forces Impact into a planning task. In this section, we discuss how to compute a solution to the plan recognition problem  $SPPR = \langle \Pi, O, \mathcal{G}, \text{PROB} \rangle$ .

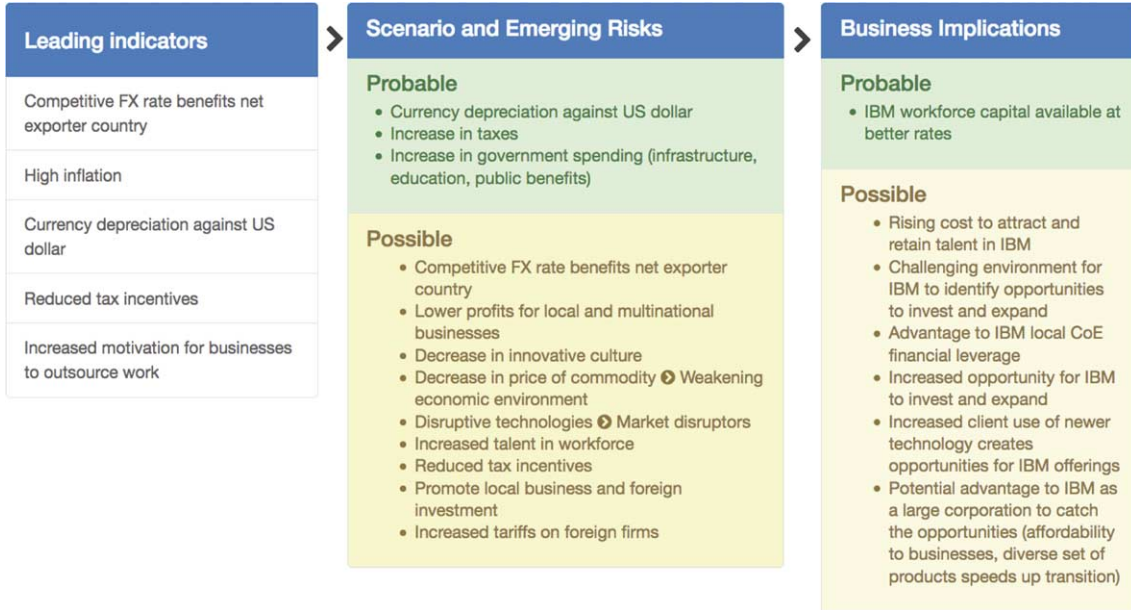


Fig. 3. Sample generated scenario.

To compute a set of scenarios (see Fig. 3 for an example of a scenario) we perform the following steps: (i) follow previous work on plan-recognition-as-planning to compile away the observations and ensure that at least one goal is satisfied, (ii) compute a set of high-quality plans on the transformed planning problem, and (iii) cluster the resulting plans into scenarios so that similar plans are grouped together. The scenarios are clusters of high-quality plans that include a trajectory of cause-effect transitions from the Mind Maps, explaining the largest possible subset of observations, and such that each plan ends with a business implication.

To transform the plan recognition problem *SPPR* into a planning task, we follow the previous work [15], which adds a set of “explain” and “discard” actions for each observation. It is important to note that the domain knowledge can be incomplete and the observations can be unreliable and not all of them explainable. Hence, the ability to discard some observations may be crucial to the solvability of the planning task. To encourage the planner to generate plans that explain as many observations as possible, a penalty is set for the “discard” action in the form of a higher cost. The penalty is relative to the cost of the other actions in the domain. Note, a high discard cost may cause a planner to consider many long and unlikely paths, while a low discard may cause a planner to discard observations without trying to explain them. Hence, we pick a

middle-ground, a penalty that is five times the cost of the *next-med* action. The resulting planning task captures both the domain knowledge that is encoded in the Mind Maps and its associated weights of the edges as well as the given set of observations, and the set of possible goals, associated with the plan recognition aspect of the problem.

To compute a large set of high-quality plans on the transformed planning task, we use top-*k* planning techniques [7, 17]. Top-*k* planning is defined in as the problem of finding *k* set of plans that have the highest quality. We use the  $K^*$  algorithm [10] in the SPA system.

To cluster the plans, we apply a hierarchical clustering algorithm on the resulting plans [3]. To compare plans with each other, we consider the union of the set of states traversed by that plan. That is, we consider the set of all predicates that were true at some point along the plan. Given that the number of ground predicates (i.e.,  $F$ ) is finite, we first represent each plan through a bit array of the same size such that 1 indicates the predicate was true at some point during the execution of that plan, and 0, otherwise. To determine the Euclidean distance between two plans, we compute an exclusive or of the corresponding bit arrays and take the square root of the sum of 1 bits. Given this distance function for each pair of plans, we compute a dendrogram bottom-up using the complete-linkage clustering method [3]. The user can specify a minimum and maximum consumable number of scenarios. These settings

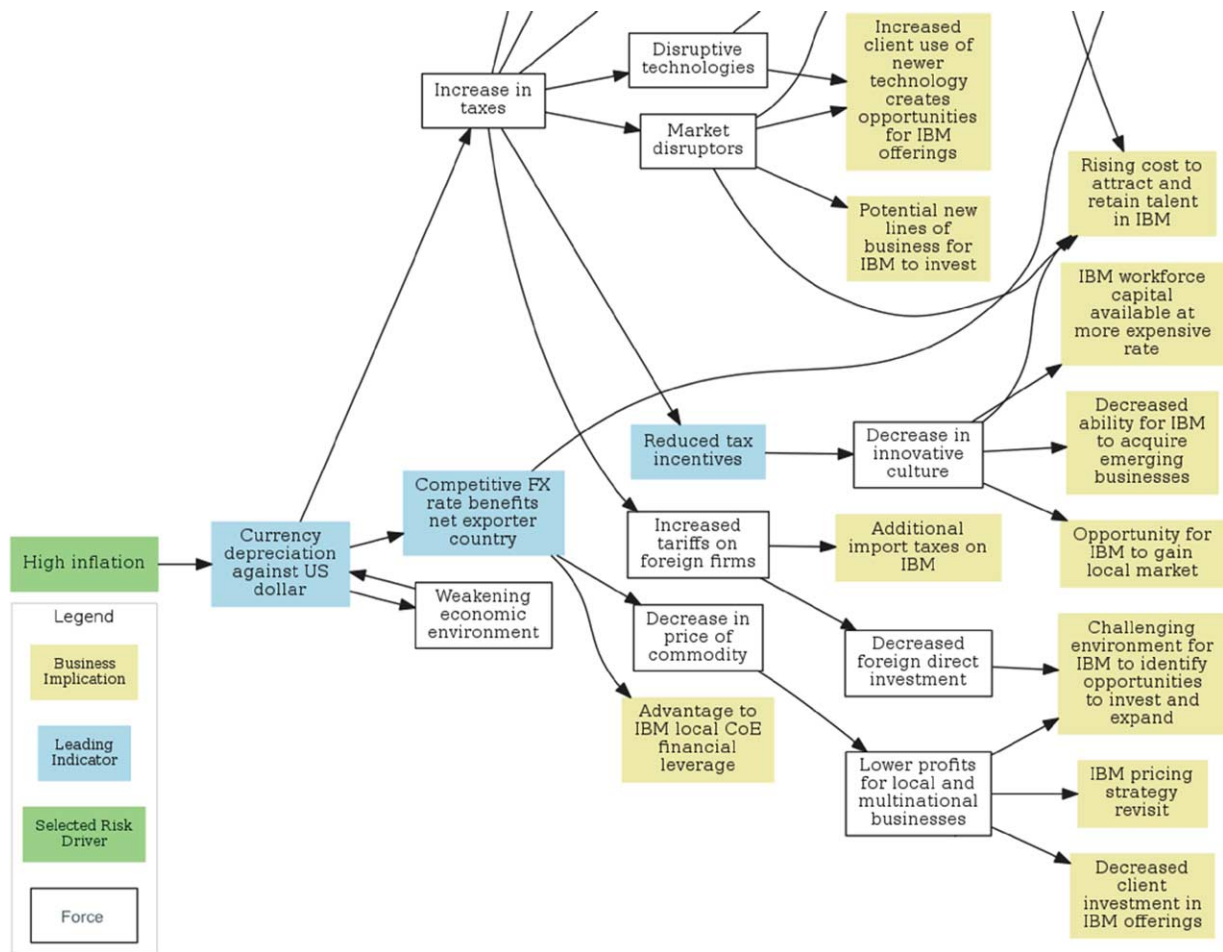


Fig. 4. Screenshot of part of an explanation graph.

are used to perform a cut through the dendrogram that yields the number of plans in the specified interval with the optimal Dunn index [4], a metric for evaluating clustering algorithms that favors tightly compact sets of clusters that are well separated. Hence, rather than presenting all plans, we group similar plans and only present 3–6 clusters of plans or scenarios to the end user.

We present the scenarios as both text summaries and graphically. We perform several tasks to prepare the scenarios for presentation. First, we separate the predicates in each cluster into business implications and regular predicates (i.e., the scenario and emerging risk). Second, we identify the leading indicators or the discriminative predicates, i.e., predicates that appear early on the plans that are part of one scenario but not other scenarios (i.e., they tend to lead to this scenario and not others); these are useful to monitor in order to determine early on whether a scenario is likely to oc-

cur. Third, we compute a summary of all plans that are part of the scenario and present this as a graph to the user (see example in Fig. 4). This serves as an explanatory tool for the predicates that are presented in each scenario. This graph also shows how the different Mind Maps are connected with each other through their shared forces.

## 8. Experimental evaluation

In this section, we evaluate the performance of the planner, quality of the clusters measured by the size of the cluster, and how informative each cluster is, measured by number of predicates and business implications. In the next section, we provide details on the pilot deployment of the Scenario Planning Advisor (SPA) tool, feedback and the lessons learned in interacting with the domain experts as well as the busi-



Table 1

Performance comparison in terms of coverage, time, and node expansion. NE is the average number of nodes expanded. time is measured in seconds. PGE/PGD is planner grounding enabled/disabled

	Blind Heuristic				LM-cut Heuristic			
	Lifted		Grounded		Lifted		Grounded	
	PGE	PGD	PGE	PGD	PGE	PGD	PGE	PGD
Solved	123	123	123	123	76	115	81	112
Time	109.95	0.90	1.19	2.11	284.13	3.58	63.05	13.90
NE	51,625	51,625	51,625	51,625	8,199	17,173	8,186	17,165

ness users. All our experiments were run on Intel(R) Xeon(R) ES-2680 @ 2.93 GHz machines, using a single core for each run, with the time and memory limit of 30 min and 2 GB, respectively.

Since the problem is computationally hard, and there exist no dedicated solver for the problem, we use the top- $k$  planner [13]. The planner is based on a heuristic search algorithm  $K^*$  [1] and implements the LM-cut heuristic [6]. It can also be run with the planner grounding step being disabled. However, this has a negative effect on the informativeness of the heuristic in use.

We create four sets of planning tasks. The first one is created using the full set of available Mind Maps (670 transitions overall) and a full set of 112 possible goals. The second one is created by taking a subset of Mind Maps, resulting in 403 overall transitions and 65 possible goals. To estimate the grounding influence on the overall performance, the last two sets mirror the first two, but are (partially) pre-grounded. We refer to these four sets as “lifted all”, “lifted small”, “grounded all”, and “grounded small”, respectively. To control the task difficulty, we vary the number of observations that are chosen randomly from the set of possible observations. For each number of observations chosen, we create 10 instances with that number of observations.

To explore the best planner configuration, we compare the planning performance of the two methods of translating the Mind Maps as well as the use of a heuristic and planner grounding. We use “lifted small” and “grounded small” with both the blind and the LM-cut heuristic [6], and with and without planner grounding. We use 10 problems of each observation set size, up to 45 observations, resulting in 150 problems overall. The timeout was set to 30 minutes. The summary of the results is shown in Table 1. Average time and node expansion are computed only on problems solved by all eight configurations.

The results show that while the use of LM-cut leads to exploring fewer nodes in search, especially with planner grounding enabled, the reduction in search ef-

fort does not compensate for the high computation time. Thus, the planner performance worsened, leading to solving fewer problems. Comparing the “lifted” to the “grounded” formulation, the heuristic informativeness does not sufficiently improve when shifting to a partially grounded representation and not enforcing a full grounding by the planner. When a grounding is enforced by the planner, the heuristic greatly reduces the number of node expansions, but even such dramatic reduction is not sufficient to compensate for the considerably increased computation time. Thus, in what follows, we restrict our attention to the lifted representation and to the blind heuristic, without enforcing full grounding by the planner.

Next, we present the evaluation of SPA performance on “lifted small” and “lifted all”. The results are shown in Table 2. The objective of this experiment is to show how the planning task size influences the performance and the resulting clusters. All entries show averages over 10 tasks of the same size. We use the same numbers of observations for both methods. The columns present the planner performance in seconds, number of observations, “Obs”, number of unexplained/discarded observations in the optimal plan, “Disc”, number of actions in the optimal plan, “Act”, and number of scenarios generated “Scen”. We also show the average and standard deviation for the number of *members* of each cluster, number of *predicates*, and number of business implications, “Bis goals”, in each scenario. The timeout was set to 30 minutes. Problems with 30 or more observations had timeouts and are not reported here.

The results show that planner performance depends not only on the Mind Maps size, but also on the number of observations. Further, as the number of observations grow, not only the planner’s run-time performance worsens, but also the number of scenarios increase, and the number of plans in the scenario decrease. On the other hand, as the number of observations increase, the number of predicates in a scenario and the number of business implications decrease, but not consistently; moreover, the low standard deviation

Table 2  
Performance comparison as we increase the number of observations and the number of Mind Maps

#Obs	Lifted Small										Lifted All									
	Time (sec)	Average Number of			Members		Predicates		Bis goals		Time (sec)	Average Number of			Members		Predicates		Bis goals	
		Disc	Act	Scen	Avg	Std	Avg	Std	Avg	Std		Disc	Act	Scen	Avg	Std	Avg	Std	Avg	Std
1	0.01	0.0	4.7	2.70	105.10	92.6	12.00	2.9	4.70	1.1	0.02	0.0	4.7	2.00	125.00	80.9	11.10	2.4	4.80	1.5
2	0.01	0.5	6.7	3.00	102.40	90.9	11.70	3.0	3.70	1.3	0.02	0.3	7.0	2.70	100.00	68.4	11.30	2.7	5.00	1.6
4	0.02	1.6	10.7	2.90	99.60	88.0	12.50	3.2	3.40	1.3	0.04	1.5	10.8	2.30	114.60	87.7	10.60	1.9	4.90	1.0
8	0.10	4.4	16.7	4.20	75.50	64.0	10.80	2.6	2.60	1.0	0.15	3.7	18.0	3.40	85.40	75.4	9.90	2.1	4.60	1.4
10	0.22	5.0	22.5	5.10	56.90	63.3	8.50	2.0	2.70	1.0	0.37	5.1	20.7	4.40	70.00	60.4	8.10	2.2	3.70	1.3
12	0.48	5.9	27.4	5.20	55.60	50.9	9.80	1.5	2.10	0.9	1.09	5.4	27.7	5.20	52.90	57.5	8.70	1.6	3.40	0.6
15	1.41	8.6	30.1	5.10	56.20	53.6	11.00	1.6	2.00	0.8	2.63	8.1	30.4	4.50	67.30	54.5	10.50	1.8	4.10	1.4
18	2.59	9.9	35.1	5.20	56.30	65.3	8.70	1.3	2.00	0.9	5.44	9.4	34.9	4.90	62.30	71.6	7.70	1.6	4.10	0.8
20	22.24	11.4	39.9	5.30	55.70	54.4	9.40	1.4	1.80	0.7	65.62	10.7	40.6	4.56	63.22	43.6	11.22	1.4	3.22	0.4
23	74.66	14.5	40.2	4.80	64.10	50.3	9.30	1.4	2.00	0.6	198.28	14.4	40.8	4.63	63.00	51.9	9.25	1.6	4.63	0.9
26	88.85	16.9	46.1	5.25	57.88	58.5	9.38	1.1	2.13	1.0	236.34	17.0	43.8	5.17	51.83	39.2	7.83	1.2	3.50	1.0

indicates that the clusters are balanced and informative. Also note that, given the number of plans to cluster, cluster sizes depend on the requested maximal number of clusters, a parameter of the clustering algorithm. This parameter was set to find between two and seven clusters. Decreasing the max cluster limit, increases the member size, as well as the number of predicates and bis implications in each scenario.

## 9. Pilot deployment and user feedback

The SPA tool was evaluated in a pilot deployment with several teams of business users at IBM, whose responsibilities included risk management within their business area. For those teams, SPA was introduced together with the new scenario planning process; hence, there was no pre-automation baseline available to compare against. In addition, the functionality provided by the overall tool is not easily reproducible, due to the broad news analysis the tool performs.

The Mind Maps were developed over the course of three months by one enterprise risk management expert working with an assistant and in consultation with other experts. While Mind Maps in general can be in any form, we briefly educated the domain experts to provide structured Mind Maps as defined in Definition 1. The pilot deployment featured the set we referred to above as “lifted all”. Additionally, the end users (i.e., the analysts) provided us with a list of possible keywords, organizations of interest, key people, key topics, and were able to pick the relevant key risk drivers when we presented them with the summary of relevant news and RSS publications. Note that while the Q&A process takes some time, the domain experts had received education and guidance and were aware of the process. The domain experts also know that the result quality depends on the richness of the Mind Maps they create. We also actively work on enhancing their experience by providing several tools to assist them. For example, we proposed an approach to suggest a list of important people, organizations and sources to the domain experts using the Wikidata Query Service [18].

The tool was configured with the help of end users. In particular, configuration values were identified based on the generated results quality and the assessment by end users. Specifically, the number of plans to find, minimum and maximum number of clusters, and action costs are all configurable and were assigned by exploring various values. In addition, all the

cost of the actions, both with respect to the translation of the domain knowledge into planning as well as the associated penalty for a discard action is also configurable. While we currently set all these configurations with the help of the users, all configurable values can be learned in the future. In addition, we have tried various syntax-based distance metrics, the one presented in the paper produced the best scenarios according to the domain experts.

The teams have universally found the tool easy to use and navigate. Although no detailed feedback was collected for each scenario, the teams have reported that approximately 80% of generated scenarios had identified the implications that directly or indirectly affect the business. By design, the tool aims at helping the business users to think outside the box and is expected to generate some irrelevant scenarios, among others. Judging by the provided comments, the teams whose business is affected by frequent political, regulatory, and economic change have found the tool more useful than those operating under relatively stable conditions. In addition, the teams found the explanation graph, a visualization of a set of plans, essential to the adoption of the tool. They believe that the explanation graph “demystifies” the tool by providing them with an explanation of why they are presented with a particular scenario. This is critical for the business users or policy-makers who would be basing their decisions on the generated scenarios.

In working with the domain experts and users from the start of the pilot deployment, we learned several lessons, which can be applicable to other settings: (1) the users are interested in using AI planning techniques, but expressing their problems in PDDL or another existing formal planning language is a barrier. To overcome this, we asked the experts to provide their knowledge in the form of structured Mind Maps, which we then translated to the planning task. Further, different experts may want to work on different parts of the problem; hence, rather than having one huge Mind Map, we allow them to provide a set of Mind Maps, each of which can be developed separately, by different experts; (2) the users are interested in being presented with several scenarios rather than one, along with the explanation of each scenario. This captures the possible alternatives rather than a precise prediction, analogous to a generation of a multiple plans rather than a single (optimal) plan; (3) the users are interested in personalized scenarios, specific to their particular use case. To address that we consider the Mind Maps as a template and allow personalization of the scenarios by

incorporating additional information provided by the Force Impact. Hence, computing a set of high-quality plans for different use cases results in different set of plans, which in turn results in different scenarios.

## 10. Related work and summary

There exist a body of work on the plan recognition problem (e.g., [11,24]). However, most approaches assume that the observations are perfect, mainly because raw data is not taken as input, but analyzed and transformed into observations in pre-processing [21]. Also, plan libraries as input are mostly assumed (e.g., [5]), whereas we use planning tools. Furthermore, there is a body of work on learning the domain knowledge (e.g., [22,23]). Our focus in addressing knowledge engineering challenges was to transform one form of knowledge, expressed in Mind Maps, into another form that is accessible by automated planners, similarly to the work of [16], adapting it to scenario planning. However, learning can be beneficial in domains in which plan traces are available.

In this paper, we applied AI planning techniques to a novel application, scenario planning for enterprise risk management. We addressed knowledge engineering challenges of encoding the domain knowledge from domain experts. To this end, we designed a tool, Scenario Planning Advisor (SPA), that takes as input raw data, news and social media posts, and interacts with the business user to obtain observations. SPA also allows uploading Mind Maps, a way of expressing the domain knowledge by the domain experts, and obtains additional information based on these Mind Maps from an automatically generated questionnaire. SPA then generates scenarios by first finding a many quality plans and then clustering the found plans into a small set of clusters, to be consumable by a human user. The SPA system is in pilot deployment with business users. The feedback received so far has been positive and confirms the benefits of our approach to the scenario generation application.

## Acknowledgements

We thank Claudio Martinez, Deepak Turaga, Nagui Halim, and Edward Shay for their guidance and support. This material is based upon work supported in whole or in part with funding from the Laboratory for Analytic Sciences (LAS). Any opinions, findings, con-

clusions, or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the LAS and/or any agency or entity of the United States Government.

## References

- [1] H. Aljazzar and S. Leue, K\*: A heuristic search algorithm for finding the K shortest paths, *Artificial Intelligence (AIJ)* **175**(18) (2011), 2129–2154, ISSN 0004-3702. doi:[10.1016/j.artint.2011.07.003](https://doi.org/10.1016/j.artint.2011.07.003).
- [2] E. Avanesov, Risk management in ISO 9000 series standards, in: *International Conference on Risk Assessment and Management*, Vol. 24, 2009, p. 25.
- [3] D. Defays, An efficient algorithm for a complete link method, *Computer Journal* **20**(4) (1977), 364–366. doi:[10.1093/comjnl/20.4.364](https://doi.org/10.1093/comjnl/20.4.364).
- [4] J.C. Dunn, A fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters, *Journal of Cybernetics* **3**(3) (1973), 32–57. doi:[10.1080/01969727308546046](https://doi.org/10.1080/01969727308546046).
- [5] R.P. Goldman, C.W. Geib and C.A. Miller, A new model of plan recognition, in: *Proceedings of the 15th Conference in Uncertainty in Artificial Intelligence (UAI)*, 1999, pp. 245–254.
- [6] M. Helmert and C. Domshlak, Landmarks, critical paths and abstractions: What’s the difference anyway? in: *Proceedings of the 19th International Conference on Automated Planning and Scheduling (ICAPS)*, 2009, pp. 162–169.
- [7] M. Katz, S. Sohrabi, O. Udrea and D. Winterer, A novel iterative approach to Top-k planning, in: *Proceedings of the 28th International Conference on Automated Planning and Scheduling*, 2018.
- [8] D.V. McDermott, PDDL — The Planning Domain Definition Language, Technical Report, TR-98-003/DCS TR-1165, Yale Center for Computational Vision and Control, 1998.
- [9] G.D. Peterson, G.S. Cumming and S.R. Carpenter, Scenario planning: A tool for conservation in an uncertain world, *Conservation biology* **17**(2) (2003), 358–366. doi:[10.1046/j.1523-1739.2003.01491.x](https://doi.org/10.1046/j.1523-1739.2003.01491.x).
- [10] F. Pommerening and M. Helmert, Optimal planning for delete-free tasks with incremental LM-cut, in: *Proceedings of the 22nd International Conference on Automated Planning and Scheduling (ICAPS)*, 2012.
- [11] M. Ramirez and H. Geffner, Plan recognition as planning, in: *Proceedings of the 21st International Joint Conference on Artificial Intelligence (IJCAI)*, 2009, pp. 1778–1783.
- [12] M. Ramirez and H. Geffner, Probabilistic plan recognition using off-the-shelf classical planners, in: *Proceedings of the 24th National Conference on Artificial Intelligence (AAAI)*, 2010, pp. 1121–1126.
- [13] A. Riabov, S. Sohrabi and O. Udrea, New algorithms for the Top-K planning problem, in: *Proceedings of the Scheduling and Planning Applications WoRKshop (SPARK) at the 24th International Conference on Automated Planning and Scheduling (ICAPS)*, 2014, pp. 10–16.
- [14] P.J. Schoemaker, Scenario planning: A tool for strategic thinking, *Sloan management review* **36**(2) (1995), 25–40.

- [15] S. Sohrabi, A. Riabov and O. Udrea, Plan recognition as planning revisited, in: *Proceedings of the 25th International Joint Conference on Artificial Intelligence (IJCAI)*, 2016, pp. 3258–3264.
- [16] S. Sohrabi, A. Riabov and O. Udrea, State projection via AI planning, in: *Proceedings of the 31st Conference on Artificial Intelligence (AAAI-17)*, 2017, pp. 4611–4617.
- [17] S. Sohrabi, A. Riabov, O. Udrea and O. Hassanzadeh, Finding diverse high-quality plans for hypothesis generation, in: *Proceedings of the 22nd European Conference on Artificial Intelligence (ECAI)*, 2016, pp. 1581–1582. 106\_Sohrabi.pdf.
- [18] S. Sohrabi, A. Riabov, O. Udrea and F. Yuan, Using lightweight semantic models to assist risk management in a large enterprise, in: *Proceedings of the 16th International Semantic Web Conference — Industry Track (ISWC-17)*, 2017. Sohrabi-ISWC-17.pdf.
- [19] S. Sohrabi, A.V. Riabov, M. Katz and O. Udrea, An AI planning solution to scenario generation for enterprise risk management, in: *Proceedings of the 32nd National Conference on Artificial Intelligence (AAAI)*, 2018, pp. 160–167.
- [20] R.M. Stulz, Rethinking risk management, *Journal of applied corporate finance* **9**(3) (1996), 8–25. doi:[10.1111/j.1745-6622.1996.tb00295.x](https://doi.org/10.1111/j.1745-6622.1996.tb00295.x).
- [21] G. Sukthankar, C. Geib, H.H. Bui, D.V. Pynadath and R.P. Goldman, *Plan, Activity, and Intent Recognition*, Morgan Kaufmann, Boston, 2014.
- [22] Q. Yang, K. Wu and Y. Jiang, Learning action models from plan examples using weighted MAX-SAT, *Artificial Intelligence (AIJ)* **171**(2–3) (2007), 107–143. doi:[10.1016/j.artint.2006.11.005](https://doi.org/10.1016/j.artint.2006.11.005).
- [23] H.H. Zhuo, T. Nguyen and S. Kambhampati, Refining incomplete planning domain models through plan traces, in: *Proceedings of the 23rd International Joint Conference on Artificial Intelligence (IJCAI)*, 2013, pp. 2451–2457.
- [24] H.H. Zhuo, Q. Yang and S. Kambhampati, Action-model based multi-agent plan recognition, in: *Proceedings of the 26th Annual Conference on Neural Information Processing Systems (NIPS)*, 2012, pp. 377–385.