The Thirty-Second AAAI Conference
on Artificial Intelligence (AAAI-18)

An AI Planning Solution to Scenario Generation for Enterprise Risk Management

Shirin Sohrabi, Anton V. Riabov, Michael Katz, Octavian Udrea
IBM T.J. Watson Research Center
1101 Kitchawan Rd, Yorktown Heights, NY 10598, USA

Abstract

Scenario planning is a commonly used method by companies to develop their long-term plans. Scenario planning for risk management puts an added emphasis on identifying and managing emerging risk. While a variety of methods have been proposed for this purpose, we show that applying AI planning techniques to devise possible scenarios provides a unique advantage for scenario planning. Our system, the Scenario Planning Advisor (SPA), 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 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.

1 Introduction

Scenario planning is a commonly used method for strategic planning (Schoemaker 1995). A major benefit to scenario planning is that it helps businesses or policy-makers to layout the possible alternative futures and anticipate them (Peterson, Cumming, and Carpenter 2003). Risk management is a set of principles that focus on the outcome for risk-taking (Stulz 1996). In this paper, we introduce Scenario Planning Advisor (SPA), a decision support system to assist organizations in generating future scenarios, identifying, and managing emerging risk, a category of risks associated with the changes in the global or local economies, politics, technology, society, and others. 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. The scenarios will include the implications for the company's finances and its ability to hire; hence, enabling the company to act immediately to minimize the negative impacts.

A variety of (manual) methods and standards for risk management under different assumptions have been developed (Avanesov 2009). The approach we take in this paper is different from previous work in that we reason about currently emerging risks based on observations from the news and social media trends, and automatically produce scenarios that both describe the current situation and summarize the future possible effects of these observations or key risk drivers. 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.

Each scenario we produce highlights: (1) the potential leading indicators, the set of facts that are likely to lead to a scenario; (2) the scenario and emerging risk, the combined set of consequences in that scenario; and (3) the business implications, a subset of potential effects of that scenario that the decision-makers care about. For example, given a high inflation risk driver, economic decline followed by a decrease in government spending can be the consequences in a scenario, while decreased client investment in the company offerings is an example of a business implication. Furthermore, an increase in the cost of transportation could have been the leading indicator for that scenario. 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.

Our system, SPA, is currently in a pilot deployment in IBM. The system is used to continuously monitor news and social media to identify current trends relevant to the company, and then generate three to six scenarios. These scenarios are then used to start a risk conversation between the analysts and decision makers. Major advantages of SPA are: (1) capturing observations from news and social media; (2) capturing the domain knowledge about the risk drivers quickly and efficiently through our knowledge engineering efforts, transferable to many other domains and applications; (3) reasoning with incomplete and biased input while mitigating the bias in generated scenarios.

The main contributions of this paper are (i) characterization of the scenario planning problem for enterprise risk management through its corresponding plan recognition problem, (ii) transformation of the domain knowledge as captured by Mind Maps into an AI planning task, (iii) computation of scenarios by applying the plan-recognition-as-planning technique generating multiple high-quality plans, and clustering them into scenarios, and (iv) evaluation of our system through its performance and the user feedback.

2 Scenario Planning Adviser (SPA)

The architecture for our system, Scenario Planning Adviser (SPA), is shown in Figure 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 media feeds. To this end, several text analytics are implemented in order to find the information that is relevant for a particular domain as filtered by the provided Topic Model. The Topic Model, provided by the domain expert, includes the list of important people, organization, and keywords. The result of the *News Aggregation* component is a set of relevant *key risk drivers*, 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), a graphical representation that encodes concepts and relations.

An example of a Mind Map for the currency depreciation force is shown in Figure 2. The Forces Impact, describes potential likelihoods and impact of a *cause* (i.e., concepts with an edge going into the main force) or a *consequence* (e.g., concepts with an edge going from the main force and all other cascading concepts). 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. Any force described by the Forces Model can be selected as a risk driver. The solution to the *SP* problem is a set 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.

3 Preliminaries

In this section, we briefly review the necessary background on AI planning and Plan Recognition. We consider planning tasks $\Pi = \langle F, A, I, G, \textsc{cost}\rangle$ in the STRIPS formalism

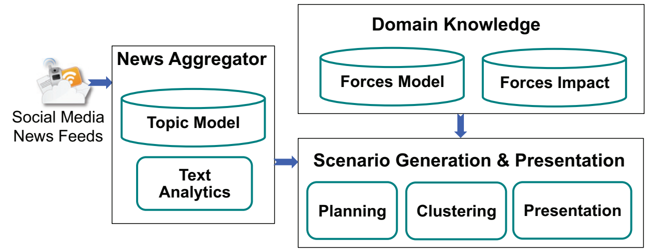

Figure 1: The SPA system architecture

extended with operator 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 Π. The state I is the *initial state* of Π. 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* $pre(a) \subseteq F$, *add* effects $add(a) \subseteq F$ and *delete* effects $del(a) \subseteq F$, and $\textsc{cost} : A \rightarrow \mathbb{R}^{0+}$ 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 $pre(a) \subseteq s$. Applying a in s results in the state $s[\![a]\!] := (s \setminus del(a)) \cup add(a)$. A sequence of actions $\pi = \langle a_1, \ldots, a_k\rangle$ is *applicable* in s if there exists a sequence of states $\langle s_0, \ldots, s_k\rangle$ 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 $s[\![\pi]\!]$. 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: $\textsc{cost}(\pi) = \sum_{i=1}^{k} \textsc{cost}(a_i)$. 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 $R = \langle \Pi, O, \mathcal{G}, \textsc{prob}\rangle$, where $\Pi = \langle F, A, I, G, \textsc{cost}\rangle$ is a planning task, $O = \{o_1, \ldots, o_m\}$, $o_i \in F$, $i \in [1, m]$ is a set of observations, $\mathcal{G} \subseteq S(\Pi)$ is the set of possible goals, and PROB is a probability distribution over the goals $\mathcal{G}$. Note, this definition includes a minor modification from previous work (Ramírez and Geffner 2009; Sohrabi, Riabov, and Udrea 2016) as it includes the planning task Π as the input to the plan recognition problem.

The solution to the PR problem is the posterior probabilities of plans that traverse the possible goals and the possible goals given observations. Plan recognition problem can be transformed to an AI planning problem and the posterior probabilities can be approximated using AI planning (e.g., (Ramírez and Geffner 2010)). Following the work of Sohrabi, Riabov, and Udrea (2016), we say that the observation is satisfied by an action sequence if it is either explained or discarded. 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.

4 Domain Knowledge

The domain knowledge in the scenario planning problem comes in two forms: Forces Model and Forces Impact. There could be many different forms and representations of these two forms of knowledge. In particular, domain experts with expertise in AI planning could use a planning language, e.g., the Planning Domain Description Language (PDDL) (McDermott 1998) to encode the Forces Model, exploiting the Forces Impact to derive costs or preferences over the actions in the domain. It is also possible to use the existing knowledge engineering tools to aid the domain experts (e.g., (Muise 2016; Simpson, Kitchin, and McCluskey 2007)).

However, as such domain experts are exceptionally rare, we anticipate the lack of proper AI planning expertise in writing the domain knowledge and the unwillingness to learn a planning language. Instead, the domain expert may choose to express her knowledge in a light-weight graphical tool such as a Mind Map and answer some simple questions that would lead to the identification of the weights of the Mind Maps edges. This knowledge can then be translated automatically to a planning language.

4.1 Forces Model as Structured Mind Map

We represent the Forces Model as a set of Mind Maps. Mind Maps can be created in a graphical tool such as FreeMind (freemind.sourceforge.net). Two example Mind Maps are shown in Figure 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 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 Figure 2(b).

Next, we define Mind Maps formally. 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.

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

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

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

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

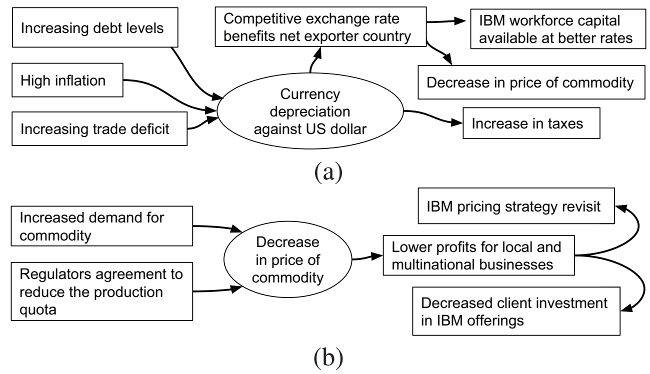

Figure 2: Part of the Mind Maps for (a) currency depreciation against US dollar (b) decrease in price of commodity.

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 φ is a sequence of symbols $[c_1, \ldots, c_{i-1}, c_i, c_{i+1} \ldots, c_{n-1}, c_n]$, $c_1, \ldots c_{n-1} \in C$, $c_n \in B$, such that there exists a Mind Map $M = \langle \Gamma, c_i, \Theta \rangle$, $M \in \mathcal{M}$, where $[c_1, \ldots, c_{i-1}] \in \Gamma$, and $[c_{i+1} \ldots, c_{n-1}, c_n] \in \Theta$.*

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. Note, Mind Maps can be connected through both cause and consequence sequences; that is, many main concepts 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.

4.2 Forces Impact via Questionnaire

Additional information on the Mind Maps is encoded through the Forces Impact. 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 Σ 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 Γ or Θ. We denote r^{impact} and $r^{likelihood}$ to denote the impact and likelihood of that pair (i.e., edge in a Mind Map). Also, we denote, $M^{importance}$, 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.

5 Transformation to Planning

The scenario planning problem, as described in Section 2, is NP-hard. This can be shown by a reduction from, e.g., the Hamiltonian Path problem. 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 (i.e., Forces Impact and Forces Model) into the planning task.

Definition 3 *Given a scenario planning problem, SP, as described in Section 2, a corresponding plan recognition problem is defined as a tuple SPPR* $= \langle \Pi, O, \mathcal{G} \rangle$, *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 uniform probability distribution over the possible goals.*

Given the corresponding plan recognition problem *SPPR*, we follow the plan-recognition-as-planning approach (Sohrabi, Riabov, and Udrea 2016) 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* π*: (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 6. 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 2.

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* $M^{importance}$*,* $M \in \mathcal{M}$*, set of all possible pairs of symbols* Σ*, and their impact and likelihood levels,* r^{impact}*,* $r^{likelihood}$*,* $r \in \Sigma$*, 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^{impact}*,* $r^{likelihood}$ *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* $M^{importance}$*.*
- A *is the union of the following action sets:*
  - $A_{next-low}$*, 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^{impact}*,* $r^{likelihood}$*,*
  - $A_{next-med}$*,* $A_{next-high}$*, similar to* $A_{next-low}$*, where low is replaced with med and high respectively,*
  - $A_{nextbis}$*, 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^{impact}*,* $r^{likelihood}$*,*
  - $A_{indicator-low}$*, for each causal sequence* $[c_1, \ldots, c_n]$ *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* $M^{importance}$*,*
  - $A_{indicator-med}$*,* $A_{indicator-high}$ *where low is replaced with med and high respectively, and*
  - $A_{achieve-goal}$ *for each* $c \in B$*, with precondition (bis* c*), add effect (achieved), no delete effect, and* 0 *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* Π *as defined above, if* φ *is a valid path for* $\mathcal{M}$*, then we can construct a sequence of actions* π*, such that* π *is a plan for the planning task* Π*. On the other hand, if* π *is a plan for the planning task* Π*, then there exists a valid path* φ *for* $\mathcal{M}$*, where* φ *can be constructed from* π*. Furthermore, a valid path* φ_1 *has a higher quality than a valid path* φ_2 *if and only if* $\text{COST}(\pi_1) < \text{COST}(\pi_2)$ *for the corresponding plans* π_1 *and* π_2*.*

Proof Sketch: ($\Rightarrow$) Given a valid path $\varphi = \langle c_1, \ldots, c_{i-1}, c_i, c_{i+1} \ldots, c_{n-1}, c_n \rangle$, we construct a plan π for the planning task Π starting with an *indicator* action for $\langle c_1, \ldots, c_{i-1}, c_i \rangle$, followed by 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), and then an *achieve-goal* action for the business implication $c_n \in B$.
($\Leftarrow$) Given a plan π for the planning task Π, we construct a valid path for $\mathcal{M}$, considering the arguments of the actions.

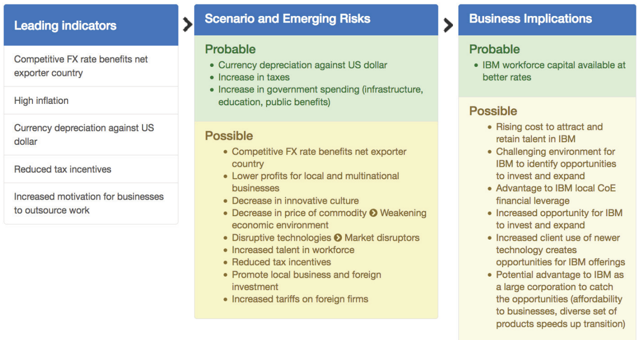

Figure 3: The screenshot of a sample generated scenario.

We also must make sure that the cost of the actions corresponds to the importance, impact and likelihood values. □

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 (McDermott 1998) 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. We evaluate the performance of both methods in the experimental evaluation section.

6 Computation of Scenarios

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* $= \langle \Pi, O, \mathcal{G} \rangle$.

To compute a set of scenarios (see Figure 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. Next, we briefly discuss each step.

To transform the plan recognition problem *SPPR* into a planning task, we follow the work of Sohrabi, Riabov, and Udrea (2016), 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

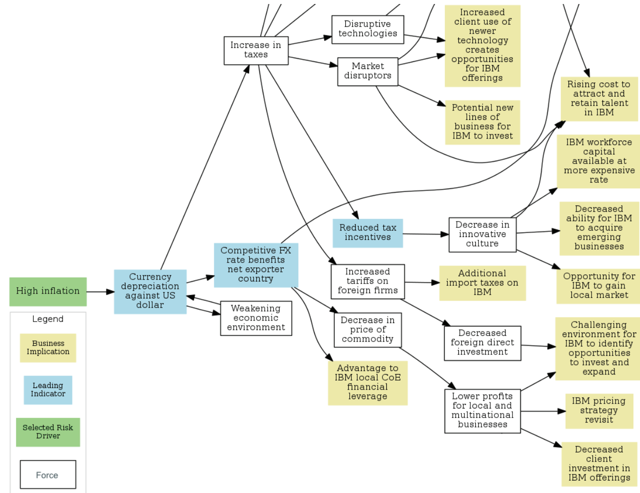

Figure 4: Screenshot of part of an explanation graph.

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 a top-k planner (Riabov, Sohrabi, and Udrea 2014). We elaborate on the top-k planner in the next section. To cluster the plans, we apply a hierarchical clustering algorithm on the resulting plans (Defays 1977). 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 exlusive 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 (Defays 1977). The user can specify a minimum and maximum consumable number of scenarios. These settings are used to perform a cut through the dendrogram that yields the number of plans in the specified interval with the optimal Dunn index (Dunn 1973), a metric for evaluat-

	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

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.

ing clustering algorithms that favors tightly compact sets of clusters that are well separated.

After we compute the set of scenarios, we automatically 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 occur. Third, we compute a summary of all plans that are part of the scenario and present this as a graph to the user. 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. Figure 3 shows a sample generated scenario and Figure 4 is the explanation graph for this scenario.

7 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 Adviser (SPA) tool, feedback and the lessons learned in interacting with the domain experts as well as the business users. All our experiments were run on a 16-core 2.93 GHz Intel(R) Xeon(R) ES-2680 processor with 264 GB RAM.

As mentioned above, we use the top-k planner (Riabov, Sohrabi, and Udrea 2014). To the best of our knowledge, it is the only available planner for the top-k planning problem, the problem of finding many top-quality plans. The planner is based on a heuristic search algorithm K^* (Aljazzar and Leue 2011) and implements the LM-cut heuristic (Helmert and Domshlak 2009). 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 (Helmert and Domshlak 2009), 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 result 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 effort 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 is 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 indicates that the clusters are balanced and informative. Also note that, given the number of plans to cluster, cluster sizes

	Lifted Small										Lifted All									
#Obs	Time	Average Number of			Members		Predicates		Bis goals		Time	Average Number of			Members		Predicates		Bis goals	
	(sec)	Disc	Act	Scen	Avg	Std	Avg	Std	Avg	Std	(sec)	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

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

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.

8 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. We also actively work on enhancing their experience by providing several tools to assist them. For example, we recently proposed an approach to suggest a list of important people, organizations and sources to the domain experts using the Wikidata Query Service (Sohrabi et al. 2017).

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 were assigned by exploring various values. 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.

9 Related Work and Summary

There exist a body of work on the plan recognition problem (e.g., (Ramírez and Geffner 2009; Zhuo, Yang, and

Kambhampati 2012)). 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 (Sukthankar et al. 2014). Also, plan libraries as input are mostly assumed (e.g., (Goldman, Geib, and Miller 1999)), whereas we use planning tools. Furthermore, there is a body of work on learning the domain knowledge (e.g., (Yang, Wu, and Jiang 2007; Zhuo, Nguyen, and Kambhampati 2013)). 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 Sohrabi, Riabov, and Udrea (2017), 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 Adviser (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.

In the future, we intend to support durations on Mind Map transitions. Another aspect is conjunctions and disjunctions between outgoing transitions, that might correspond to multiple effects and non-deterministic effects, respectively.

10 Acknowledgements

We thank Claudio Martinez, Fang Yuan, and Finn McCoole for providing the domain expertise. We thank 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, conclusions, or recommendations expressed in this material are those of the authors and do not reflect the views of the LAS and/or any agency or entity of the United States Government.

References

Aljazzar, H., and Leue, S. 2011. K*: A heuristic search algorithm for finding the k shortest paths. *Artificial Intelligence* 175(18):2129–2154.

Avanesov, E. 2009. Risk management in ISO 9000 series standards. In *International Conference on Risk Assessment and Management*, volume 24, 25.

Defays, D. 1977. An efficient algorithm for a complete link method. *Computer Journal* 20(4):364–366.

Dunn, J. C. 1973. A fuzzy relative of the isodata process and its use in detecting compact well-separated clusters. *Journal of Cybernetics* 3(3):32–57.

Goldman, R. P.; Geib, C. W.; and Miller, C. A. 1999. A new model of plan recognition. In *Proceedings of the 15th Conference in Uncertainty in Artificial Intelligence (UAI)*, 245–254.

Helmert, M., and Domshlak, C. 2009. Landmarks, critical paths and abstractions: What's the difference anyway? In *Proceedings of the 19th International Conference on Automated Planning and Scheduling (ICAPS)*, 162–169.

McDermott, D. V. 1998. PDDL — The Planning Domain Definition Language. Technical Report TR-98-003/DCS TR-1165, Yale Center for Computational Vision and Control.

Muise, C. 2016. Planning.Domains. In *the 26th International Conference on Automated Planning and Scheduling - Demonstrations*.

Peterson, G. D.; Cumming, G. S.; and Carpenter, S. R. 2003. Scenario planning: a tool for conservation in an uncertain world. *Conservation biology* 17(2):358–366.

Ramírez, M., and Geffner, H. 2009. Plan recognition as planning. In *Proceedings of the 21st International Joint Conference on Artificial Intelligence (IJCAI)*, 1778–1783.

Ramírez, M., and Geffner, H. 2010. Probabilistic plan recognition using off-the-shelf classical planners. In *Proceedings of the 24th National Conference on Artificial Intelligence (AAAI)*.

Riabov, A.; Sohrabi, S.; and Udrea, O. 2014. 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)*, 10–16.

Schoemaker, P. J. 1995. Scenario planning: a tool for strategic thinking. *Sloan management review* 36(2):25.

Simpson, R. M.; Kitchin, D. E.; and McCluskey, T. L. 2007. Planning domain definition using GIPO. *Knowledge Eng. Review* 22(2):117–134.

Sohrabi, S.; Riabov, A.; Udrea, O.; and Yuan, F. 2017. 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)*.

Sohrabi, S.; Riabov, A.; and Udrea, O. 2016. Plan recognition as planning revisited. In *Proceedings of the 25th International Joint Conference on Artificial Intelligence (IJCAI)*.

Sohrabi, S.; Riabov, A.; and Udrea, O. 2017. State projection via AI Planning. In *Proceedings of the 31st Conference on Artificial Intelligence (AAAI-17)*.

Stulz, R. M. 1996. Rethinking risk management. *Journal of applied corporate finance* 9(3):8–25.

Sukthankar, G.; Geib, C.; Bui, H. H.; Pynadath, D. V.; and Goldman, R. P. 2014. *Plan, Activity, and Intent Recognition*. Boston: Morgan Kaufmann.

Yang, Q.; Wu, K.; and Jiang, Y. 2007. Learning action models from plan examples using weighted max-sat. *Artificial Intelligence* 171(2-3):107–143.

Zhuo, H. H.; Nguyen, T.; and Kambhampati, S. 2013. Refining incomplete planning domain models through plan traces. In *Proceedings of the 23rd International Joint Conference on Artificial Intelligence (IJCAI)*, 2451–2457.

Zhuo, H. H.; Yang, Q.; and Kambhampati, S. 2012. Action-model based multi-agent plan recognition. In *Proceedings of the 26th Annual Conference on Neural Information Processing Systems (NIPS)*, 377–385.