Planning-based Scenario Generation for Enterprise Risk Management

Shirin Sohrabi and **Anton V. Riabov** and **Octavian Udrea**
IBM T.J. Watson Research Center
1101 Kitchawan Rd, Yorktown Heights, NY 10598, USA
{ssohrab, riabov, udrea}@us.ibm.com

Abstract

Scenario planning is a commonly used method that various organizations use to develop their long term plans. Scenario planning for risk management puts an added emphasis on identifying the extreme yet possible risks that are not usually considered in daily operations. While a variety of methods and tools have been proposed for this purpose, we show that formulating an AI planning problem, and applying AI planning techniques to develop the scenarios provides a unique advantage for scenario planning. Our system, the Scenario Planning Advisor (SPA), takes as input the relevant news and social media trends that characterize the current situation, where a subset of them is selected to represent key observations, as well as the domain knowledge. The domain knowledge is acquired using a graphical tool, and then automatically translated to a planning domain. We use a planner to generate multiple plans explaining the observations and projecting future states. The resulting plans are clustered and summarized to generate the scenarios for use in scenario planning. We discuss our knowledge engineering methodology, lessons learned, and the feedback received from the pilot deployment of the SPA system in a large international company. We also show our experiments that measure planning performance and how balanced and informative the generated scenarios are as we increase the complexity of the problem.

1 Introduction

Scenario planning is a commonly used method for strategic planning (Schoemaker 1995). Scenario planning involves analyzing the relationship between forces such as social, technical, economic, environmental, and political trends in order to explain the current situation in addition to providing insights about the future. A major benefit to scenario planning is that it helps businesses or policy-makers learn about the possible alternative futures and anticipate them. While the expected scenarios are interesting for verification purposes, scenarios that are surprising to the users (e.g., policy-makers businesses) are the ones that are the most important and significant (Peterson *et al.* 2003).

Risk management is a set of principles that focus on the outcome for risk-taking (Stulz 1996). A variety of methods and standards for risk management under different assumptions have been developed (Avanesov 2009). In this paper, we address scenario planning for risk management, the problem of generating scenarios with a significant focus on identifying the extreme yet possible risks that are not usually considered in daily operations. The approach we take in this paper 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 not to find a precise answer, that is to predict or forecast, but rather to project the possible alternative scenarios that may need consideration. 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, and the *business implications*, a subset of potential effects of that scenario that the users (e.g., policy-makers, businesses) care about. The business implications are akin to the set of possible goals.

For example, given a high inflation observation, economic decline followed by a decrease in government spending can be the consequences or the possible effects in a scenario, while decreased client investment in the company offerings is an example of a business implication (i.e., the resulting goal). 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.

In this paper, we propose to view the scenario planning problem for enterprise risk management as a problem that can be translated to an AI planning problem. An intermediate step is a plan recognition problem, where the set of given business implications forms the set of possible goals, and the observations are selected from the news and social media trends. The domain knowledge is acquired from the domain expert via a graphical tool and is then automatically translated to an AI planing domain. AI planning is in turn used to address the plan recognition problem (Ramírez and Geffner 2009; Sohrabi *et al.* 2016a; 2017). Top-k planning or finding a set of high-quality plans is used to generate multiple plans that can be grouped into a scenario (Riabov *et al.* 2014; Sohrabi *et al.* 2016b). The set of plans is then clustered and summarized to generate the scenarios. Hence, each scenario is a collection of plans that explain the observations and considers the possible cascading effects of the actions to identify potential future outcomes.

2 System Architecture

The system 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* component and the *Domain Knowledge* component. 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 observations, a subset of which can be selected by the business user and is fed into the *Scenario Generation* 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 effects 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 Mind Map (`http://freemind.sourceforge.net/wiki/`), a graphical tool that encodes concepts and relations. An example of a Mind Map for the currency depreciation force is shown in Figure 3. The Forces Impact, describes potential likelihoods and impact of a cause (i.e., concepts with an edge going into the main force) or an effect (e.g., concepts with an edge going from the main force and all other cascading concepts). The *Scenario Generation* component takes the domain knowledge and the key observations and automatically generates a planning problem whose outcome when clustered in the post-processing step generates a set of alternative scenarios.

Our system is currently deployed for an international organization. We use a company name Acme, for anonymity, in our examples. The system generates thousand plans and presents three to six scenarios to the business user. The extensive feedback we have collected has been encouraging and helpful in improving our system. We report on our knowledge engineering efforts, collected feedback, and the lessons learned in the rest of this paper.

3 Problem Definition

In this section, we briefly review necessary background on AI planning and Plan Recognition before defining the scenario planning for risk management problem.

Definition 1 *A planning problem is a tuple* $P = (F, A, I, G)$*, where* F *is a finite set of fluent symbols,* A *is a set of actions with preconditions,* $\text{PRE}(a)$*, add effects,* $\text{ADD}(a)$*, delete effects,* $\text{DEL}(a)$*, and action costs,* $\text{COST}(a)$*,* $I \subseteq F$ *defines the initial state, and* $G \subseteq F$ *defines the goal state.*

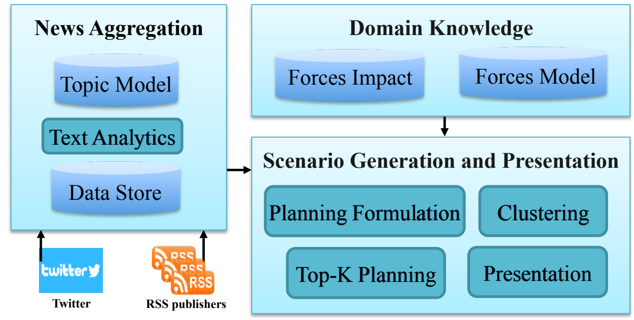

Figure 1: The SPA system architecture

The solution to the planning problem, P, is a sequence of executable actions, $\pi = [a_0, ..., a_n]$ such that if executable from the initial state, I, meets the goal (i.e., $G \subseteq \delta(a_n, \delta(a_{n-1}, \ldots, \delta(a_0, I)))$, where $\delta(a, s) = ((s \backslash \text{DEL}(a)) \cup \text{ADD}(a))$ defines the successor state.

Definition 2 *A Plan Recognition (PR) problem is a tuple* $R = (F, A, I, O, \mathcal{G}, \text{PROB})$*, where* (F, A, I) *is the planning domain as defined above,* $O = \{o_1, ..., o_m\}$*, where* $o_i \in F$*,* $i \in [1, m]$ *is the set of (partially ordered) observations,* $\mathcal{G}$ *is the set of possible goals* G*,* $G \subseteq F$*, and* PROB *is a probability distribution over* $\mathcal{G}$*,* $P(G)$*.*

The solution to the PR problem is the posterior probabilities $P(\pi|O)$ and $P(G|O)$. Plan recognition problem can be transformed to an AI planning problem and the posterior probabilities can be approximated using AI planning (Ramírez and Geffner 2010; Sohrabi *et al.* 2016a). Note, the observations are said to be satisfied by an action sequence if it is either explained or discarded following the work of Sohrabi et al. 2016a. 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.

Definition 3 *A scenario planning for enterprise risk management problem is defined as a tuple* $SP = (F, A, I, O, \mathcal{G})$*, where* (F, A, I) *is the planning domain acquired by the domain experts,* $O = \{o_1, ..., o_m\}$*, where* $o_i \in F$*,* $i \in [1, m]$ *is a set of observations selected from the news and social media trends,* $\mathcal{G}$ *is a set of possible goals* $G \subseteq F$*; the set of goals are called business implications in the scenario planning problem.*

As shown in Figure 1, the input to the SPA system are raw social media posts and news articles with RSS feeds. The News Aggregation component analyzes such news and posts and suggests possible observations. In the deployment of the SPA system, we addressed unordered set of observations as input; however, in theory, the observations can be expressed in any Linear Temporal Logic (LTL) formula (Sohrabi *et al.* 2011).

The solution to the SP problem is defined as a set of scenarios, where each scenario is a collection of plans Π such that: (1) each plan $\pi = [a_0, ..., a_i, a_{i+1}, ..., a_n]$ is an action

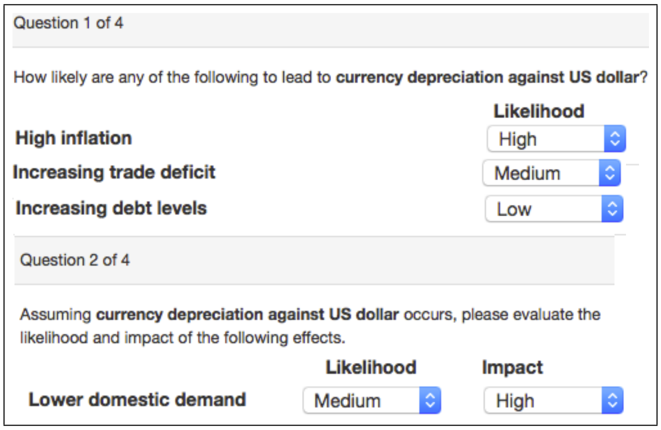

Figure 2: Sample questions

sequence that is executable from the initial state I and results in state $s = \delta(a_n, \ldots, \delta(a_0, I))$, (2) at least one of the goals is met (i.e., $\exists G \in \mathcal{G}$, where $G \subseteq s$), and (3) the set of observations is satisfied by the action sequence $[a_0, ..., a_i]$ (i.e., observations are either explained or discarded). The SP problem can be thought of as a plan recognition problem, where observations and a set of goals are given. Rather than computing $P(\pi|O)$ and $P(G|O)$, the solution to the SP problem is a set of scenarios showcasing the alternative possible outcomes.

4 Knowledge Engineering

While several knowledge engineering tools exists, most of them assume that the domain expert has some AI planning background and these tools provide the additional support in writing the domain knowledge (e.g., (Muise 2016; Simpson *et al.* 2007)). However, 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 their knowledge in a light-weight graphical tool and have this knowledge translated automatically to a planning language such as Planning Domain Description Language (PDDL) (McDermott 1998). In this section, we discuss the representation of the domain knowledge and its translation to planning.

As shown in Figure 1, the domain knowledge comes in two forms: Forces Model and Forces Impact. Forces Model, is the domain knowledge corresponding to the causes and effects of the different forces influencing the risks in a business organization such as the economy, currency, corruption, social unrest, and taxes. The domain experts express these relationship for each force trends (e.g., economic decline and economic growth) in separate Mind Maps. A Mind Map[1] is a graphical method that can be used to express the Forces Model in a simple way. The Mind Maps can be created in a tool such as FreeMind[2] which produces an XML representation of the Mind Maps which can serve as an input to our

[1] `https://en.wikipedia.org/wiki/Mind_map`
[2] `http://freemind.sourceforge.net/wiki/`

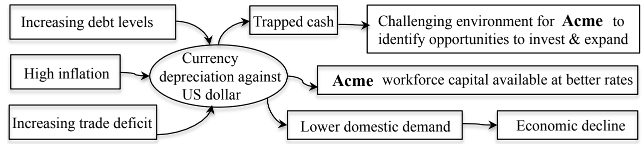

Figure 3: Part of the Mind Map for the currency depreciation against US dollar force.

system. An example Mind Map is shown in Figure 3. The force in this Mind Map is the currency depreciation. The concepts with an edge going towards the force, are the possible causes, and the concepts with an outgoing edge from the force, are the possible effects. The causes and effects can appear in chains, and cascade to other causes, and effects, with a leaf concept of either a business implication (i.e., the planning goal), or another force, with its own separate Mind Map that describes it. For example, "Acme workforce capital available at better rates" is an example of a business implication, where Acme is the name of the organization. Note, one of the leafs of this Mind Map, economic decline, is another force which would be described in a separated Mind Map. Any of the concepts in the Mind Map, except for the business effects, can serve as observations in order to generate the scenarios.

Additional information on the Mind Maps is encoded through the Forces Impact, which is captured by a series of automatically generated questions based on the Mind Maps. These questions are created by a script that reads the XML encoding of the Mind Maps. Sample questions are shown in Figure 2. The domain expert is given options of low, medium, and high in addition to the option of "do not know" in which a default value is selected for them. The answers to these questions determine the weight of the edges in the Mind maps.

The domain knowledge encoded in the Mind Maps (i.e., Forces Model), together with the answers from the questionnaire (i.e., Forces Impact), is automatically translated into a planning language such as PDDL. There are at least two ways to translate the Mind Maps into a planning language. The first method, we call "ungrounded", defines one general and ungrounded set of actions in the PDDL domain file with many possible groundings of the actions based on the given Mind Maps. The domain file includes an action named "indicator" for each of the causes in a Mind Map. There would be three different "indicator" actions, one for each level (i.e., "indicator-low", "indicator-med" and "indicator-high"). The levels are determined based on the answers to the questionnaire. The domain file also includes an action named "next", and "next-bis" for each of the edges in the Mind Map. The "next" action also has three different versions, one for each level. The "next-bis" actions do not have levels and are those that end in a business implication concept (i.e., a concept that includes the name of the company).

Table 1 shows part of the planning domain. For example, the "next-med" action will be grounded by setting the parameter x1 to "increasing trade deficit" and the parameter x2 to the "currency depreciation against US dollar". Each

```
(:action next-med
 :parameters (?x1 - occ ?x2 - occ)
 :precondition (and (occur ?x1)
                    (next-med ?x1 ?x2))
 :effect (and (occur ?x2)
              (not (occur ?x1))
              (increase (total-cost) 10)))

(:action indicator-med
 :parameters (?y - force ?x - occ)
 :precondition (and (indicator-med ?y ?x))
 :effect (and (occur ?x)
              (increase (total-cost) 15)))

(:action next-bis
 :parameters (?x1 - occ ?x2 - bisimplication)
 :precondition (and (occur ?x1)
                    (next-bis  ?x1 ?x2)
 :effect (and (bis-implication-achieved)
              (increase (total-cost) 6)))
```

Table 1: Part of the planning domain.

of the "next" actions (-low, -med, -high) have a cost that maps to the importance of that edge such that lower impact/likelihood answers map to a higher cost. Hence, while the domain is fixed, based on the answers obtained by the domain experts, the actions will have a different set of possible groundings defined in the problem file. The "next-bis" action is the action that if executed, indicates that at least one of the business effects have been reached and the "bis-implication-achieved" predicate is set to true; this is the goal of the planning problem. The problem file (i.e., the initial state) will include all the possible groundings of these actions by including a grounding for the predicates "(next-med ?from ?to)", "(next-bis ?from ?to)", and "(indicator-med ?y ?x)". Note that the size of the Mind Map leads to a larger problem file, as the domain file is fixed. A successful plan maps to an execution of an "indicator" action, followed by the execution of one or more "next" actions, followed by an execution of a "next-bis" action. This maps to a path through the connected Mind Maps.

The second method to translate the Mind Maps into a planning language is called "grounded" which as the name suggests, defines one action per each edge in the Mind Map in addition to one action for each of the causes in the Mind Map in the planning domain itself. So rather than having one fixed planning domain which can get grounded by the problem file, the second approach fully specifies all the possible actions in the planning domain. We evaluate the performance of both methods in the experimental evaluation.

5 Computing Plans

In the previous section, we discussed how to translate the information available in the Mind Maps into a planning domain and problem. However, we are also given the set of observations as the input and we need to compile away the observations in order to use planning. To do so we follow the work of Sohrabi et al. 2013; 2016a which adds a set of "explain" and "discard" actions for each observation. The discard action can be selected in order to leave some observations unexplained. The observations are driven from news and social media posts and not all of them are reliable; in addition, some of them could be mutually exclusive and not all of them could be explainable. Hence, it is important to have the ability to discard some observations. However, 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 cost. The penalty is relative to the cost of the other action in the domain; we currently set it to be five times the cost of a "next-med" action. After considering multiple options, this seemed to be good a middle-ground option between the two extremes; a high discard cost will cause the planner to consider many long and unlikely paths, while a low discard will cause the planner to discard observations without trying to explain them. In addition, to ensure all observations are considered, whether explained or discarded, a set of special predicates, one per each observation is used and must hold true for each of the "next-bis" actions. This ensures that a plan that meets one of the goals also has considered all of the observations. To disallow different permutation of the discard action, we discard observations using a fixed order.

The resulting planning problem 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 possible set of goals, associated with the plan recognition aspect of the problem. To compute a set of high-quality plans on the transformed planning problem, we use the top-k planning approach proposed in (Riabov *et al.* 2014; Sohrabi *et al.* 2016b). Top-k planning is defined in as the problem of finding k set of plans that have the highest quality. The best known algorithm to compute the set of top-k plans is based on the k shortest paths algorithm called K^* (Aljazzar and Leue 2011) which also allows use of heuristics search. We use the K^* algorithm together with the LM-cut heuristic (Pommerening and Helmert 2012) in our system. Next, we discuss how the generated plans are post-processed into the scenarios.

6 Computing Scenarios

To compute the type of scenarios shown in Figure 4, we perform a set of post-processing steps on the computed set of plans. All of the post-processing steps are done automatically. First, we identify the number of plans out of the top-k plans (e.g., 1000) generated by the planner to consider for scenario generation. We argue that this number is problem-dependent rather than being a fixed number for all problems. To calculate the cost cutoff, we calculate the average and the standard deviation of the cost of all plans among the top-k plans. We then consider plans that have a lower cost than the average cost subtracted by the standard deviation. The number of plans considered for scenario generation is shown under the "# of Plans" column in Table 2.

Next, we cluster the resulting plans to create scenarios. Hence, rather than presenting all plans, we group similar plans and only present 3-6 clusters of plans to the end user. We cluster plans according to the predicates present in the last state. Given that the number of ground predicates (i.e, $\mathcal{F}$) is finite, we first represent each plan through a bit array of the same size such that 1 indicates the predicate is in the

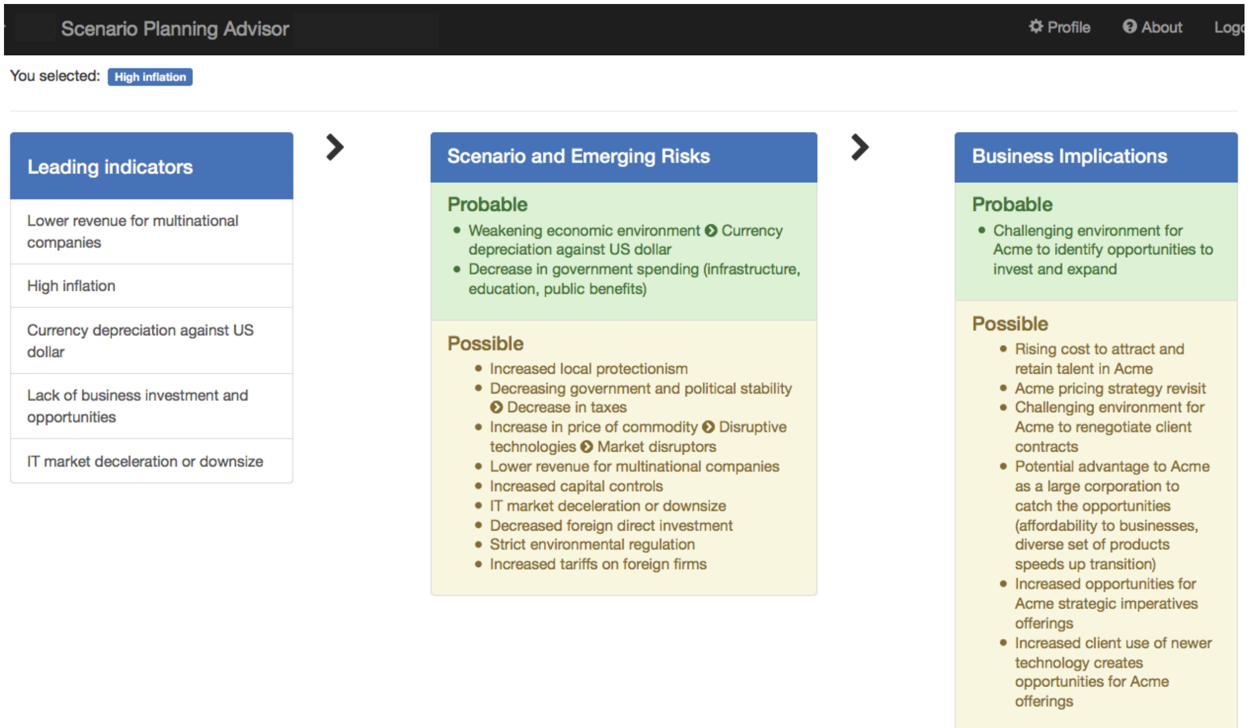

Figure 4: The screenshot of a sample generated scenario for the high inflation observation. Each scenario is divided into three parts, the leading indicators, scenario and emerging risks, and the business implications.

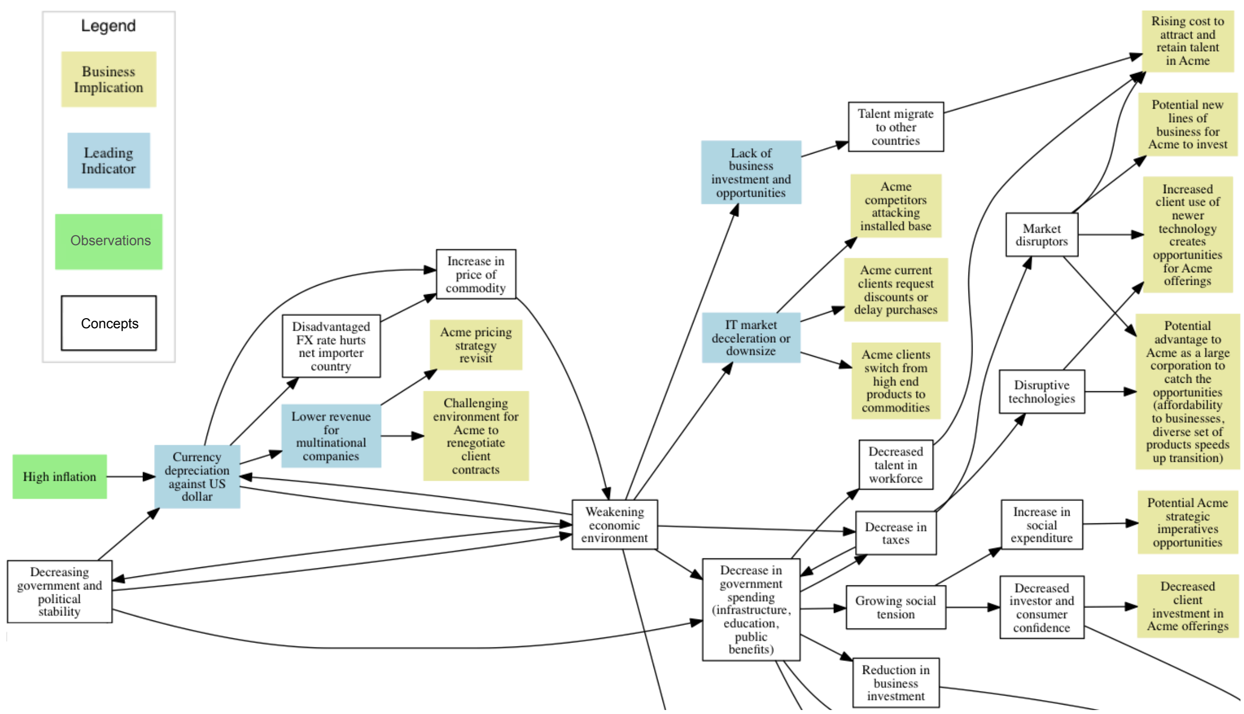

Figure 5: Part of the screenshot of a explanation graph for the scenario shown in Figure 4. Observations are shown in green, leading indicators are shown in blue, and business implications are shown in yellow.

final state, and 0 indicates that the predicate is not in the final state. To determine the Euclidean distance between two plans, we compute a XOR of the corresponding bit arrays and take the square root of the sum of 1 bits. Normally, we want to avoid plans with opposite predicates (e.g., weakening/strengthening economic environment, increase/decrease in inflation, etc.) ending up in the same cluster. To ensure this, we add a penalty factor to the number of 1 bits we use to compute the distance for every pair of opposite predicates. 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 evaluating clustering algorithms that favors tightly compact sets of clusters that are well separated.

After post-processing is complete, we automatically perform several tasks to prepare the scenarios for presentation. First, we separate the predicates in each cluster (scenario) into business implications and regular predicates. At the same time, we separate probable and possible predicates in each of these categories by determine the proportion of plans where the predicate is present in the last state from all plans in the scenario; predicates that appear in more than 66% of plans are put into the probable category, those that appear between 25% and 66% are placed in the possible category. Second, we identify 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. Figure 5 shows an example of this graph. 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 concepts that are shared between them.

7 Experimental Evaluations

In this section, we evaluate: (1) the performance of the planner, (2) quality of the clusters measured by the size of the cluster, and (3) how informative each cluster 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 2.5 GHz Intel Core i7 processor with 16 GB RAM.

We compare the performance of the planner on our two proposed methods to translate the Mind Maps into a planning domain: "ungrounded" and "grounded". The "grounded" method creates 670 actions when considering the full set of Mind Maps. We remove some of these Mind Maps creating 403 actions instead and report on that result under the "ungrounded small" method. To increase the difficulty of the problem, we increase the size of the O. Observations are chosen randomly from the set of possible observations.

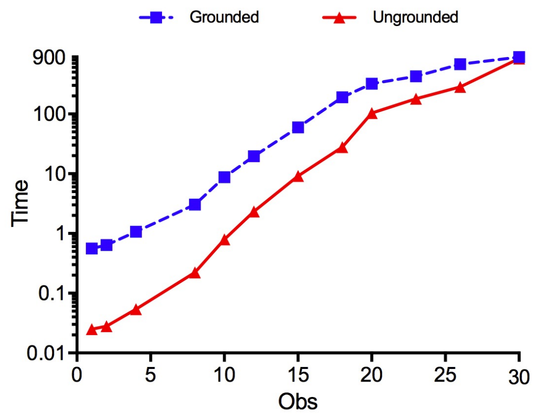

Figure 6: Planning performance comparison between the "grounded" and "ungrounded" methods, as we increase the number of observations. The time is in seconds and is shown in logarithmic scale.

Table 2 presents a comparison between "ungrounded small" and "grounded". The objective of this experiment is to show how the planning domain size influence performance and the generated clusters. All numbers shown in each row are averages over 10 runs of the same type of problem, where the same number of observations is considered in both cases. The columns show the planning performance in seconds, total number of business implications, $\mathcal{G}$, number of actions, $\mathcal{A}$, number of observations O, number of discarded observations in the optimal plan, "# of Discards", number of plans considered for scenario generation, "# of Plans", and number of scenarios generated "# of Scenarios". We also show the average, standard deviation, max, and min count on the number of members of each cluster, number of predicates, and number of business implications in each scenario. We used a timeout of 900 seconds. The problems with 30 or more observations did not finish within the time limit.

The results show that the performance of the planner depends on both the number of observations and the size of the domain, as expected. As the number of observations grow the planner's performance worsens but this does not influence the number of plans, the number of scenarios, size of the clusters, or the number of scenario predicates. However, the number of business implications decreases, as expected, as the observation size grows. Looking at the average number of cluster members, the average number of scenarios predicates, and the average number of bossiness implications, the results show that the clusters in both cases are balanced and informative.

We also compare the planning performance between two methods of translating the Mind Maps. The results in logarithmic scale is shown in Figure 6. Each shown point in the figure is an average over 20 instances. The results show that in our current implementation, as the number of observations increases, planning performance using

	Time	$\|\mathcal{G}\|$	$\|\mathcal{A}\|$	$\|O\|$	#of Discards	#of Plans	#of Scenarios	Cluster Members Avg	Cluster Members σ	Cluster Members Max	Cluster Members Min	Scenario Predicates Avg	Scenario Predicates σ	Scenario Predicates Max	Scenario Predicates Min	Bis Implications Avg	Bis Implications σ	Bis Implications Max	Bis Implications Min
Ungrounded Small	0.03	65	403	1	0.0	129.0	3.8	37.0	28.6	76.9	11.2	9.6	2.7	13.5	6.6	4.8	1.7	7.1	2.7
	0.03	65	403	2	0.5	141.7	3.8	39.9	31.7	83.4	6.5	9.8	3.1	13.7	5.4	4.1	1.7	6.1	1.8
	0.05	65	403	4	1.6	120.5	3.6	34.6	27.9	72.1	7.9	10.9	2.7	13.9	6.9	3.7	1.2	5.3	2.1
	0.22	65	403	8	4.4	122.4	3.8	34.8	33.4	82.6	4.3	10.0	2.4	13.0	6.9	2.1	0.9	3.5	1.3
	0.80	65	403	10	5.0	112.6	4.5	25.6	26.0	71.5	5.6	7.6	2.0	10.1	5.4	2.3	0.8	3.8	1.6
	2.33	65	403	12	5.9	100.1	4.2	25.3	20.7	56.2	4.4	9.4	1.4	11.1	7.4	1.7	0.4	2.6	1.2
	9.16	65	403	15	8.8	104.8	3.9	30.2	25.6	68.5	8.8	10.6	1.2	12.4	8.8	1.9	0.4	2.8	1.5
	27.85	65	403	18	9.9	92.8	4.8	20.2	23.5	61.3	3.0	8.5	1.2	10.3	6.7	1.6	0.5	2.4	1.3
	103.71	65	403	20	11.3	117.7	3.9	30.9	26.8	68.0	3.7	9.0	1.4	11.0	7.3	1.8	0.6	2.5	1.0
	179.90	65	403	23	14.9	103.7	4.1	26.3	21.2	58.6	4.4	9.0	1.4	11.1	6.9	1.9	0.6	2.7	1.2
	282.87	65	403	26	16.9	90.6	4.9	20.3	19.0	53.5	5.3	9.5	1.1	11.3	7.8	1.6	0.3	2.0	1.3
Ungrounded	0.03	112	670	1	0.0	91.5	4.4	24.4	16.6	48.6	6.6	7.0	2.5	10.4	4.3	4.5	1.7	6.6	2.2
	0.04	112	670	2	0.4	132.1	4.3	34.4	32.2	80.3	3.7	8.0	3.0	11.7	4.0	3.8	1.8	6.1	1.5
	0.08	112	670	4	1.5	114.1	3.6	32.9	30.7	77.9	4.3	10.2	2.9	13.2	6.1	3.6	1.4	6.0	2.3
	0.35	112	670	8	3.7	109.7	3.6	31.5	24.6	65.9	7.0	9.1	1.9	11.4	6.5	3.8	1.3	5.4	2.3
	1.17	112	670	10	5.1	139.4	4.2	34.6	27.9	73.2	5.9	7.8	2.0	10.1	4.9	2.6	0.8	3.9	1.6
	3.35	112	670	12	5.4	99.5	4.8	22.8	24.8	64.7	3.5	8.6	1.9	11.0	6.3	1.6	0.4	2.8	1.2
	22.01	112	670	15	8.1	92.3	4.1	23.3	22.1	57.9	3.2	9.9	1.8	12.0	7.0	2.5	1.0	4.0	1.5
	85.73	112	670	18	9.4	88.5	4.3	22.2	19.2	51.6	6.7	7.2	1.1	8.7	5.5	2.2	0.3	2.7	1.7
	144.89	112	670	20	10.7	124.3	5.1	26.0	19.4	57.0	5.0	9.0	1.0	10.0	7.2	2.1	0.2	2.3	1.9
	284.73	112	670	23	14.5	106.8	4.8	24.5	23.9	62.5	4.0	8.6	1.6	10.6	6.5	2.9	0.6	3.6	2.0
	511.95	112	670	26	16.8	80.0	4.7	17.2	9.0	30.2	7.8	7.8	1.0	9.5	6.5	1.7	0.7	2.8	1.2

Table 2: Comparison between "ungrounded" and "ungrounded small" as we increase the number of observations: "grounded" considers all of the Mind Maps, "ungrounded small" considers a smaller set of Mind Maps.

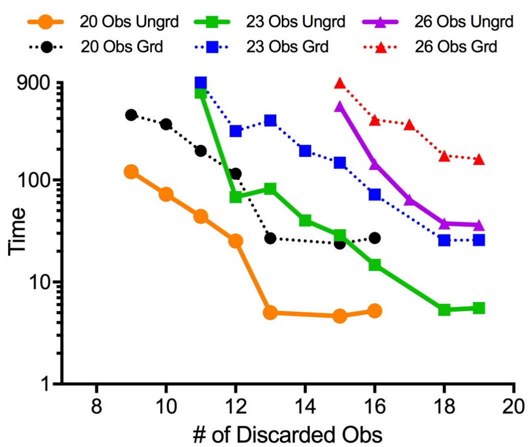

Figure 7: Planning performance comparison between two methods (i.e, "grounded" and "ungrounded") as the number of discarded observations in an optimal plan increases when considering problems with 20, 23, and 26 observations. The time is in seconds and is shown in logarithmic scale.

the "ungrounded" method is significantly better than the "grounded" method.

Considering problems with 20, 23, and 26 observations, we also looked at the number of discarded observations in the optimal plan in each case. This indicates whether or not the observations are explainable in a single path through the Mind Maps. The results in logarithmic scale is shown in Figure 7. The results confirm that the performance of the "ungrounded" method is better than the "grounded" method. It also shows that as the number of discarded observation increases, the planning time decreases. This seems to indicate that the planner identifies the unexplainable observations, through its heuristics, and does not spend time on explaining the unexplainable observations.

Based on these results, we conclude that performance of the planner depends on number of observations, the size of the domain, the method used in the translation of the Mind Maps, as well as the number of unexplainable observations. Given this result, we deployed the "ungrounded" method and use the full set of Mind Maps.

8 User Experience

The SPA tool was evaluated in a pilot deployment with 7 teams of business users, 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 tool cannot be reproduced manually due to the broad news analysis the tool provides.

The Mind Map 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 expert to provide Mind Maps that have one force (e.g., currency deprecation against US dollar) as their main concept and provide causes and consequences of this force in one Mind Map; the concepts with an edge to the central concept and the concepts with an edge from the main concept and their cascading effects where the last effect is either a business implication or another force with its own separate Mind Map. This ensures that we can automatically translate the Mind Maps into a planning language. We used 23 Mind Maps in the pilot deployment and used the "ungrounded" method to translate the Mind Maps. The resulting planning problem that aggregates the knowledge of all Mind Maps (i.e., the grounding of the actions based on the edges on the Mind Maps) has around 350 predicates and 670 actions.

Additionally, the end users (i.e., the analysts) provided us with a list of possible keywords, such as organizations of interest, key people, key topics, and were able to pick the relevant sequence of observations when we presented them with the summary of relevant news and RSS publications. For RSS publications, around 3,000 news abstracts from 64 publishers, and for Twitter, around 73,000 tweets from around 32,000 users matched our keyword search criteria.

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 implications directly or indirectly affecting the business. By design, the tool is trying to help the business users to think outside the box and it is expected that some of the scenarios it generates will not be relevant. 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, visualization of a set of plans, essential to the adaptation of the tool (Figure 5). 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.

The suggestions for improvement focused primarily on the need for further automated assistance in selecting observations based on the news, to ensure that no important context is lost, and on the additional information about the scenarios. Several teams have requested confidence levels or at least ranking information provided with the generated scenarios. We believe this is an interesting future direction and believe more accurate models are required in order to provide that additional information.

In working with the domain experts and users from the start of the pilot deployment, we learned several lessons: (1) 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; (2) The users wanted personalized scenarios specific to their particular use case. To address that we consider the Mind Maps as a template that holds true for all use cases and allow personalization of the scenarios by incorporating different weights of the edges of the Mind Maps. As mentioned previously we automatically generate a serious of questions in order to obtain the impact and likelihoods that are specific to a use case. 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; (3) The domain experts found themselves continuously updating the Mind Maps after interacting with the tool and we had to enable those continuous updates. In addition to building the automated technique of translating the Mind Maps to planning language, we assigned unique identifiers to each of the concepts in the Mind Maps. This allowed us to develop scripts for supervised detection and propagation of the associated knowledge throughout theses changes.

9 Related Work and Summary

There exist a body of work on the plan recognition problem with several different approaches (e.g., (Zhuo *et al.* 2012)). However, most approaches assume that the observations are perfect, mainly because they do not take as input the raw data and that they do not have to analyze and transform the raw data into observations (Sukthankar *et al.* 2014). Also, most plan recognition approaches assume plan libraries are given as input, whereas we use AI planning (Goldman *et al.* 1999). Furthermore, there is a body of work on learning the domain knowledge (Yang *et al.* 2007; Zhuo *et al.* 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 planners. Learning can be beneficial in domains in which plan traces are available.

In this paper, we applied AI planning techniques for a novel application, scenario planning for enterprise risk management and addressed knowledge engineering challenges of encoding the domain knowledge from domain experts. To this end, we designed Scenario Planning Adviser (SPA), that takes as input the raw data, news and social media posts, and interacts with the business user to obtain key observations. SPA also allows upload of Mind Maps, as one way of expressing the domain knowledge by the domain experts, and obtains additional information based on these Mind Maps by an automatically generated questionnaire. SPA then automatically generates scenarios by first generating large number of plans and then clustering the generated plans into a small set (i.e., 3-6) in order to be consumable by a human user. The SPA system is in pilot deployment with several teams of business users. The feedback we have received so far have been positive and show that our approach seems promising for this application.

10 Acknowledgements

We thank Fang Yuan and Finn McCoole at IBM for providing the domain expertise. We thank Nagui Halim and Edward Shay for their guidance and support. We also thank our LAS collaborators. 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 necessarily reflect the views of the LAS and/or any agency or entity of the United States Government.

References

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

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

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

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

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

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

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

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

Florian Pommerening and Malte 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.

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

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

Anton Riabov, Shirin Sohrabi, and Octavian 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)*, pages 10–16, 2014.

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

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

Shirin Sohrabi, Jorge A. Baier, and Sheila A. McIlraith. Preferred explanations: Theory and generation via planning. In *Proceedings of the 25th National Conference on Artificial Intelligence (AAAI)*, pages 261–267, 2011.

Shirin Sohrabi, Octavian Udrea, and Anton Riabov. Hypothesis exploration for malware detection using planning. In *Proceedings of the 27th National Conference on Artificial Intelligence (AAAI)*, pages 883–889, 2013.

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

Shirin Sohrabi, Anton Riabov, Octavian Udrea, and Oktie Hassanzadeh. Finding diverse high-quality plans for hypothesis generation. In *Proceedings of the 22nd European Conference on Artificial Intelligence (ECAI)*, 2016.

Shirin Sohrabi, Anton Riabov, and Octavian Udrea. State projection via ai planning. In *Proceedings of the 31st Conference on Artificial Intelligence (AAAI-17)*, 2017.

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

Gita Sukthankar, Christopher Geib, Hung Hai Bui, David V. Pynadath, and Robert P. Goldman. *Plan, Activity, and Intent Recognition*. Morgan Kaufmann, Boston, 2014.

Qiang Yang, Kangheng Wu, and Yunfei Jiang. Learning action models from plan examples using weighted max-sat. *Artificial Intelligence*, 171(2-3):107–143, February 2007.

Hankz Hankui Zhuo, Qiang Yang, and Subbarao Kambhampati. Action-model based multi-agent plan recognition. In *Proceedings of the 26th Annual Conference on Neural Information Processing Systems (NIPS)*, pages 377–385, 2012.

Hankz Hankui Zhuo, Tuan Nguyen, and Subbarao Kambhampati. Refining incomplete planning domain models through plan traces. In *Proceedings of the 23rd International Joint Conference on Artificial Intelligence (IJCAI)*, pages 2451–2457, 2013.