

RAE: AN AGENT FOR MODELING HUMAN DECISION PROBLEMS AS POMDPs

by

SOHAN M. NIPUNAGE

(Under the Direction of Prashant Doshi)

ABSTRACT

Most of the decision aid agents are designed for specific domain problems. These agents are based on different approaches based on the specific needs of the decision problem. Our research is on developing a decision-aid system as an agent called 'Rae' which will attempt to represent any decision problem as a POMDP in an intuitive way. We model the approach of solving any decision problem through a decision-making agent designed to help users elicit their decision problems. We present approaches for elicitation of user preferences into reward function of POMDP based on Analytical hierarchical process, symmetry of actions, elicitation of transitions, elicitation of observations, policy representation and designing an interactive text/voice based agent for POMDP. We present hypotheses of a decision making agent and evaluate Rae by conducting a human usability study.

INDEX WORDS: Decision-aid, POMDP, Decision making, Interactive decision making agent, Agent modeling, Preference elicitation, Symmetry, Rational policy, Analytical hierarchical process

RAE: AN AGENT FOR MODELING HUMAN DECISION PROBLEMS AS POMDPs

by

SOHAN M. NIPUNAGE

B.E., University of Mumbai, 2015

A Thesis Submitted to the Graduate Faculty
of The University of Georgia in Partial Fulfillment
of the
Requirements for the Degree

MASTER OF SCIENCE

ATHENS, GEORGIA

2019

© 2019

Sohan M. Nipunage

All Rights Reserved

RAE: AN AGENT FOR MODELING HUMAN DECISION PROBLEMS AS POMDPs

by

SOHAN M. NIPUNAGE

Approved:

Major Professor: Prashant Doshi

Committee: Adam Goodie
John Miller

Electronic Version Approved:

Suzanne Barbour
Dean of the Graduate School
The University of Georgia
August 2019

DEDICATION

To Madhukar and Prabodhini, my caring grandparents, Shubhangi and Milind, my loving parents, Priyanka, my supportive sister, Paris, my supportive brother and my friends.

ACKNOWLEDGMENTS

I would like to thank my major professor, Dr. Prashant Doshi, for his supervision, advice and guidance.

I'm grateful to Dr. Adam Goodie from the department of Psychology at University of Georgia for his immense help crucial contributions for the Cognitive studies, IRB application, and patiently answering to all my questions, sometimes- basic questions.

I would like to thank Layton Hayes and Keyang He of University of Georgia for helping me proofread my thesis and providing valuable feedback.

I would like to thank Dnyanada Shirsat, Ameya Limaye and all my friends for always being supportive to my decisions in curriculum and their advice in research studies.

I would like to thank my parents, sister and grandparents for always supporting me throughout my ups and downs and helping me build myself the best in everything I come across.

TABLE OF CONTENTS

	Page
ACKNOWLEDGMENTS	v
LIST OF FIGURES	viii
LIST OF TABLES	xii
CHAPTER	
1 INTRODUCTION	1
1.1 PROBLEM DESCRIPTION	2
1.2 BACKGROUND	3
1.3 CONTRIBUTIONS	13
1.4 THESIS OUTLINE	15
2 RELATED WORK	16
2.1 INTRODUCTION	16
2.2 ECONOMIC AGENTS	16
2.3 PARK MANAGER AGENTS	19
2.4 ARTIFICIAL DECISION MAKING FOR INTELLIGENT BUILDINGS	23
2.5 AI FRAMEWORK FOR CLINICAL DECISION-MAKING	25
2.6 DXPLAIN: DIAGNOSTIC DECISION-SUPPORT SYSTEM	28
2.7 ACS NSQIP SURGICAL RISK CALCULATOR	31
2.8 DECISION AIDS FOR PATIENTS	34
3 RATIONAL DECISION MAKING AGENT	38
3.1 VOICE/TEXT AGENT	38

3.2	PREFERENCE ELICITATION	40
3.3	ELICITATION OF TRANSITIONS	49
3.4	ELICITATION OF OBSERVATIONS	57
3.5	INTERACTIVE POLICY	63
4	EXPERIMENTAL ANALYSIS	67
4.1	HYPOTHESES	67
4.2	HUMAN USABILITY STUDY	81
4.3	DATA ANALYSIS	83
5	CONCLUSION	86
5.1	LIMITATIONS AND FUTURE WORK	88
	BIBLIOGRAPHY	90
	APPENDIX	
A	SURVEY 1	94
B	SURVEY 2	97

LIST OF FIGURES

1.1	Agent interacting with it’s environment	4
1.2	Generic Hierarchic Process	7
1.3	Format for pairwise comparisons	7
1.4	Encoding the tiger problem as a vertex-colored graph. Two vertices have the same color if and only if their shapes and fillings are the same.	12
2.1	Sim Parc general Architecture by Briot et al. (2008) [19]	20
2.2	Park manager decision process by Briot et al. (2008) [19]	21
2.3	Architecture of MDP for clinical decision making [22]	25
2.4	A dynamic decision network for clinical decision-making [22].	27
3.1	Model of Rae, as seen by the user before exploring any of the parameters . .	39
3.2	Model of Rae, after entering state and action labels into the model.	40
3.3	Model of Rae, after entering state and action labels into the model, it asks users to enter preferences of actions to each state.	41
3.4	Model of Rae, after entering preferred actions for each state into the model. .	42
3.5	Architecture of generating reward matrix from user preferences	48
3.6	Model of Rae after removing invalid transitions for action Listen (L) of the tiger domain problem.	50
3.7	Rae suggests that action OL and OR could be symmetric.	52
3.8	User enters the chances of transitions they chose and symmetry function helps to fill in values for symmetric actions. In this case, when user enters values for action OL, Rae fills values for action OR simultaneously.	53
3.9	Rae generates transition matrix in POMDP format and associates chance probabilities entered by user into transition values.	53

3.10	Architecture of eliciting transition function of the domain problem	56
3.11	User enters observation labels and chooses an action that gives observations.	57
3.12	State labels entered through which the user observes the specified observation variables	59
3.13	Observation matrix for only single states through which the user observes the observation-variables	59
3.14	Multiple state labels entered through which the user observes the specified observation variables	60
3.15	Observation matrix for multiple states through which the user observes the observation-variables	60
3.16	Architecture of eliciting observations of the domain problem.	62
3.17	Rational policy suggested by Rae based on user inputs. This is an interactive policy it has suggested the first action Listen (L).	63
3.18	After clicking action L, policy asks to choose what we observe.	64
3.19	Choosing observation 'Growl from the right' (GR), policy suggests next action which is L.	64
3.20	Choosing action L after the policy recommendations.	64
3.21	After the user chooses action L in the previous policy, he is asked to choose one of the observations.	65
3.22	Architecture of interactive policy	65
4.1	This graph is based on scores received from participants based on satisfaction of working and interacting with Rae in Survey: 2 as described in Appendix B	68
4.2	This graph is based on scores received from participants based on interaction with Rae and their scores about effectiveness based on a question in Survey: 2 as described in Appendix B	68
4.3	This analysis is based on relation of the problem domain with their own based on a question in Survey: 2 as described in Appendix B	69

4.4 Survey-1 (State): Analysis of data we collected from Survey questionnaire -1 which asks participants to select state from the available choices. 70

4.5 Survey-1 (Action): Analysis of data we collected from Survey questionnaire -1 which asks participants to select action from the available choices. 70

4.6 Survey-1 (Preference) is based on the understanding of preferences that are asked to generate reward matrix. 70

4.7 Survey-2 (Preference) is based on the question asked in survey questionnaire-2 which asks whether the participants believe their preferences of actions should have been different. 70

4.8 Survey-1 (Transitions) is based on the understanding transitions. 71

4.9 Survey-2 (Transitions) is based on the question asked in survey questionnaire-2 which asks whether the participants believe their transitions should have been different. 71

4.10 Survey-1 (Observation) is based analysis of the question asked about choosing observations before exploring the policy. 72

4.11 Survey-2 (Observation) is based analysis of the question asked about choosing observations after exploring the policy. 72

4.12 This analysis is based on analysis of policy which tells us about participants’ satisfaction with the recommended policy by Rae in Survey: 2 as described in Appendix B 72

4.13 This analysis is based on analysis of policy which tells us about participants’ satisfaction with the recommended policy by Rae in Survey: 2 as described in Appendix B 73

4.14 This analysis is based on the question asked in Survey-2 as described in Appendix B regarding probability judgment of the participant. 74

4.15 Survey: 1, Q: 7, Relation of ‘Change of Major problem’ to their personal decision problem. 77

4.16	Survey-2, Q:1, Satisfaction of the recommended policy.	78
4.17	Survey-2 (Policy Satisfaction) is based analysis of policy which tells us about participants' satisfaction with the recommended policy.	83
4.18	Survey-2 is based on analysis of how helpful were the Rae's replies and aid during the process until policy exploration.	84
4.19	This analysis is based on analysis of satisfaction of working with Rae.	84
4.20	This analysis is based on relation of the problem domain with their own.	85

LIST OF TABLES

1.1	Gradation scale for quantitative comparison of alternatives.	8
1.2	Transition probabilities of the tiger problem	11
1.3	Observation probabilities of the tiger problem	11
1.4	Reward function of the tiger problem	11
3.1	Reward Function of Tiger Problem	45
4.1	Table representing Pearson's Bi-variate one-tailed correlations calculated in IBM SPSS software between the survey question and mean,median & modal values of the transition probabilities	76
4.2	Table representing Pearson's Bi-variate one-tailed correlations calculated in IBM SPSS software between the relation of problem domain and policy satisfaction.	78

CHAPTER 1

INTRODUCTION

Artificial intelligence researchers have always been interested in developing intelligent decision aids with applications ranging from critical financial, medical, and logistics domains to low-stakes processes, such as product recommendations or automated software configuration. Decision theory provides solutions given the system dynamics and outcome utilities. However, user utilities are often unknown and vary more widely than decision dynamics. Since obtaining full preferences is usually infeasible, this presents a serious problem to the deployment of intelligent agents that make decisions or recommendations for users with distinct utilities. Therefore, preference elicitation emerges as one of the more important current challenges in artificial intelligence. Our research is on developing an algorithm based on Analytical Hierarchical Process to elicit user preferences into a reward function of a POMDP.

Current solution and modelling approaches to Markov Decision Processes (MDPs) scale poorly with the size of MDP. Model minimization methods address this issue by exploiting redundancy in problem specification to reduce the size of the MDP model. Symmetries in a problem specification can give rise to special forms of redundancy that are not exploited by existing minimization methods. We apply a symmetry-method which suggests about possible symmetry in the transitions as a decision aid for reducing the number of inputs required for transitions of the POMDP for their decision problem.

1.1 PROBLEM DESCRIPTION

Problems associated with decision making are the clarity of parameters required for solving a decision problem and to solve such a decision problem we need to convert it into a Partially Observable Markov Decision process which then can be solved and provided with a rational policy. But to convert any decision problem into a POMDP, we need to design a decision-aid system which can help assist anyone to have their decision problems converted into a POMDP model.

Decision making in real world is complex and is dependent on various different parameters such as human experience, goal, rewards, etc. but these parameters are difficult to give numerical values for. For example, let us consider an example of Tiger Problem Domain: Assume you are facing two closed doors, behind one of the doors is a bag of gold which you would want to have. But there is a Tiger who is moving behind the two doors and you don't know his exact location. But you can listen to his growls and plan accordingly which door to open that will give you the bag of gold instead of getting eaten by the tiger. Now, if we ask to give a numerical value to gold and getting eaten by the tiger and then link those to actions, it will be difficult and everyone will come up with numbers based on their experiences which may or may not be correct. Hence, we need a model that could help assist humans with decision problems and elicit numerical values or relations.

1.2 BACKGROUND

1.2.1 PARTIALLY OBSERVABLE MARKOV DECISION PROCESS

A sequential decision problem for a fully observable, stochastic environment with a Markovian transition model and additive rewards is called Markov decision process (MDP) and consists of a set of states, a set of ACTIONS(s) of actions in each state, a transition model $P(s'—s,a)$ and a reward function $R(s)$ [1].

A *Markov Decision Process* is a tuple $\langle S, A, \Psi, P, R \rangle$, where S is the set of states, A is the set of actions, $\Psi \subseteq S \times A$ is the set of admissible state-action pairs, $P : \Psi \times S \rightarrow [0, 1]$ is the transition probability function with $P(s, a, s')$ being the probability of transition from state s to state s' under action a , and $R : \Psi \rightarrow \mathfrak{R}$ is the expected reward function, with $R(s, a)$ being the expected reward for performing action a in state s [11]. We assume that the rewards are bounded. Let $A_s = \{a | (s, a) \in \Psi\} \subseteq A$ denote the set of actions admissible in state s . We assume that for all $s \in S$, A_s is non-empty. In this work we assume that the set of states and set of actions are finite. A *stochastic policy* π is a mapping from Ψ to the real interval $[0, 1]$ with $\sum_{a \in A_s} \pi(s, a) = 1$ for all $s \in S$. For any $(s, a) \in \Psi$, $\pi(s, a)$ gives the probability of picking action a in state s . The *value* of state s under policy π is the expected value of the discounted sum of future rewards starting from state s and following policy π thereafter. The value function V^π corresponding to a policy π is the mapping from states to their values under π . It can be shown that V^π satisfies the Bellman equation:

$$V^\pi(s) = \sum_{a \in A_s} \pi(s, a) \left[R(s, a) + \gamma \sum_{s' \in S} P(s, a, s') V^\pi(s') \right]$$

where $0 \leq \gamma \leq 1$ is a discount factor. This formulation is known as the discounted sum of rewards criterion.

The solution of an MDP is an optimal policy π^* that uniformly dominates all other possible policies for that MDP.

$$V^*(s) = \max_{a \in A_s} \sum_{s' \in S} P(s, a, s') [R(s, a) + \gamma V^*(s')]$$

The description of Markov decision processes assumed that the environment was fully observable. With this assumption the agent always knows which state it is in, which means the policy depends only on the current state. When the environment is only partially observable, the agent does not necessarily know which state it is in, so it cannot execute action $\pi(s)$ recommended for that state. The utility of a state s and the optimal action in s depend not just on s , but also on how much the agent knows when it is in s . Partially observable MDPs are usually viewed as much more difficult than ordinary MDPs. The real world is a POMDP. A POMDP has the same elements as an MDP- the transition model $P(s'|s, a)$, actions $A(s)$, and reward function $R(s)$ but it also has a sensor model $P(e|s)$ which are observations of the agent in that state. In POMDPs, the belief state becomes a probability distribution over all possible states [1].

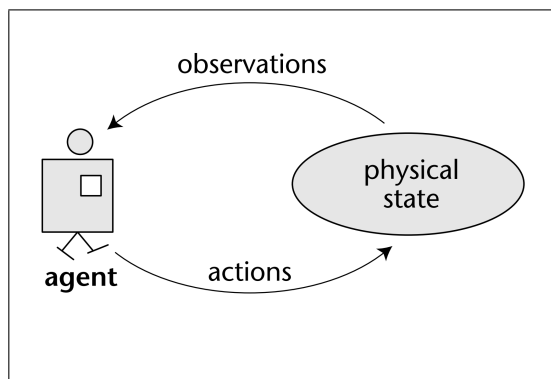


Figure 1.1: Agent interacting with it's environment

1.2.2 ANALYTICAL HIERARCHICAL PROCESS

Preference elicitation is a key problem facing the deployment of intelligent systems that make or recommend decisions on the behalf of users. Since not all aspects of a utility function have the same impact on object-level decision quality, determining which information to extract from a user is itself a sequential decision problem, balancing the amount of elicitation effort and time with decision quality. We formulate this problem as a partially-observable Markov decision process (POMDP). Because of the continuous nature of the state and action spaces of this POMDP, standard techniques cannot be used to solve it [6].

The complexity of the modern world is a much-acknowledged fact. As the human race develops, complexity increases. Technology has created various artifacts to relieve us of manual, routine and time-consuming tasks. The predictable and deterministic world of the past has been replaced by the uncertain, random and disorderly world of today. Technological advances in multiple fields of human activity have created a planet on which things happen at electronic speed. Rapidly increasing complexity and information overload have schemed together to drastically reduce the time available for making decisions. The decision-maker is stressed, overloaded with unsolicited information, has not enough time to analyze the situation, and yet must make decisions that have high-risk implications or consequences. What does the decision-maker need? Human decision-making in the world characterized above needs a quick-response analysis of the situation that somehow captures the decision-maker's intuition, judgment and experience. This can then be combined with detailed quantitative analysis based on the information glut that is churned out from the plethora of process measurements, balanced scorecards, business intelligence, data accumulation and information generation techniques [7].

Owing to its simplicity and ease of use, the AHP has found ready acceptance by busy managers and decision-makers. It helps structure the decision-maker's thoughts and can help in organizing the problem in a manner that is simple to follow and analyze. Broad areas in which the AHP has been applied include alternative selection, resource allocation, forecasting, business process re-engineering, quality function deployment, balanced scorecard, bench-marking, public policy decisions, healthcare, and many more. Basically the AHP helps in structuring the complexity, measurement and synthesis of rankings.

Now, as discussed earlier in the problem domain, we know how difficult it is to come up with values for the rewards of getting the bag of gold or getting eaten by the tiger. Also, we need to associate this reward values with the states and actions. A modified Analytical Hierarchical Process algorithm will help us provide standard values for reward function based on user preferences.

The AHP provides a means of decomposing the problem into a hierarchy of sub- problems which can more easily be comprehended and subjectively evaluated. The subjective evaluations are converted into numerical values and processed to rank each alternative on a numerical scale. The methodology of the AHP can be explained in following steps:

Step 1: The problem is decomposed into a hierarchy of goal, criteria, sub-criteria and alternatives. This is the most creative and important part of decision-making. Structuring the decision problem as a hierarchy is fundamental to the process of the AHP. Hierarchy indicates a relationship between elements of one level with those of the level immediately below. This relationship percolates down to the lowest levels of the hierarchy and in this manner every element is connected to every other one, at least in an indirect manner. A hierarchy is a more orderly form of a network. An inverted tree structure is similar to a hierarchy. Saaty suggests that a useful way to structure the hierarchy is to work down from the goal as far

as one can and then work up from the alternatives until the levels of the two processes are linked in such a way as to make comparisons possible. Figure 2.1 shows a generic hierarchic structure.

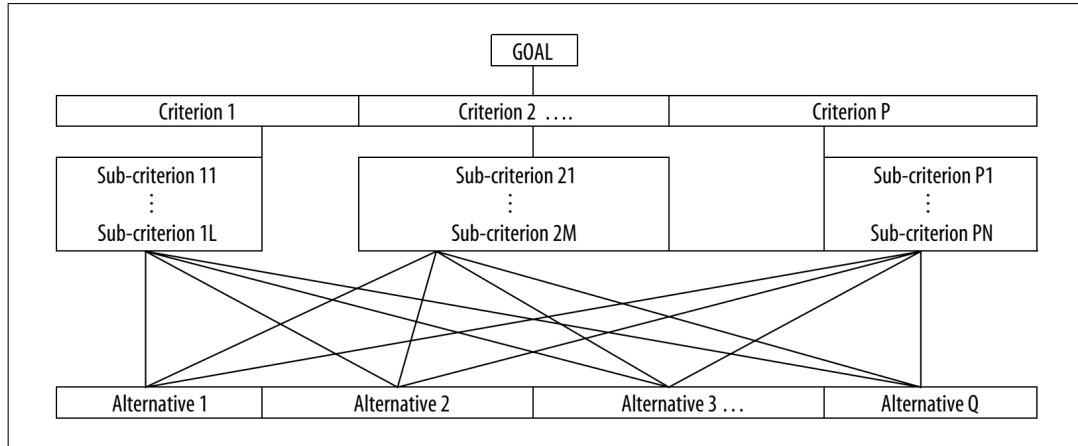


Figure 1.2: Generic Hierarchic Process

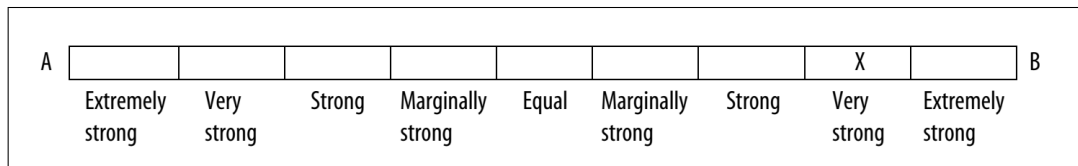


Figure 1.3: Format for pairwise comparisons

At the root of the hierarchy is the goal or objective of the problem being studied and analyzed. The leaf nodes are the alternatives to be compared. In between these two levels are various criteria and sub-criteria. It is important to note that when comparing elements at each level a decision-maker has just to compare with respect to the contribution of the lower-level elements to the upper-level one. This local concentration of the decision-maker on only part of the whole problem is a powerful feature of the AHP.

Step 2: Data are collected from experts or decision-makers corresponding to the hierarchic structure, in the pairwise comparison of alternatives on a qualitative scale as described below. Experts can rate the comparison as equal, marginally strong, strong, very strong, and

Table 1.1: Gradation scale for quantitative comparison of alternatives.

Option	Numerical value(s)
Equal	1
Marginally Strong	3
Strong	5
Very Strong	7
Extremely Strong	9
Intermediate values to reflect fuzzy inputs	2,4,6,8
Reflecting dominance of second alternative compared with the first	Reciprocals

extremely strong. The opinion can be collected in a specially designed format as shown in Figure 2.2. “X” in the column marked “Very strong” indicates that B is very strong compared with A in terms of the criterion on which the comparison is being made. The comparisons are made for each criterion and converted into quantitative numbers as per Table 2.3.

Step 3: The pairwise comparisons of various criteria generated at step 2 are organized into a square matrix. The diagonal elements of the matrix are 1. The criterion in the i^{th} row is better than criterion in the j^{th} column if the value of element (i, j) is more than 1; otherwise the criterion in the j^{th} column is better than that in the i^{th} row. The (j, i) element of the matrix is the reciprocal of the (i, j) element.

Step 4: The principal eigenvalue and the corresponding normalized right eigenvector of the comparison matrix give the relative importance of the various criteria being compared. The elements of the normalized eigenvector are termed weights with respect to the criteria or sub-criteria and ratings with respect to the alternatives.

Step 5: The consistency of the matrix of order n is evaluated. Comparisons made by this method are subjective and the AHP tolerates inconsistency through the amount of redundancy in the approach. If this consistency index fails to reach a required level then answers to comparisons may be re-examined. The consistency index, CI , is calculated as

$$CI = (\lambda_{max} - n)/(n - 1)$$

where λ_{max} is the maximum eigenvalue of the judgment matrix. This CI can be compared with that of a random matrix, RI . The ratio derived, CI/RI , is termed the consistency ratio, CR . Saaty suggests the value of CR should be less than 0.1.

Step 6: The rating of each alternative is multiplied by the weights of the sub-criteria and aggregated to get local ratings with respect to each criterion. The local ratings are then multiplied by the weights of the criteria and aggregated to get global ratings.

The AHP produces weight values for each alternative based on the judged importance of one alternative over another with respect to a common criterion.

1.2.3 SYMMETRIES IN MDPs

Partially observable Markov decision processes (POMDPs) are a natural model for stochastic sequential decision problems under observation uncertainty. However, due to intractability results in solving POMDPs, common algorithms are hindered by poor scalability on the size of the problem. One approach to address this issue is to exploit the redundancy present in the model: by aggregating equivalent states of the model, we can derive a minimized model which can be solved by traditional algorithms with a reduced computational complexity [10].

Markov Decision Processes (MDPs) are a popular way to model stochastic sequential decision problems. But most modeling and solution approaches to MDPs suffer from the fact that they scale poorly with the size of the problem [11]. While modeling real world scenarios, often there is a lot of redundancy in the MDP model. Model minimization methods exploit such redundancy in the problem specification to derive smaller models, i. e., models with fewer states, by aggregating "equivalent" states.

Such model minimization methods have been explored in depth for Markov decision processes (MDPs) and POMDPs by defining the notion of state equivalence and providing a model minimization algorithm that computes the partition of the state space so that the blocks of the partition can be regarded as abstract states. The result is a reduced MDP, while preserving the equivalence to the original MDP, which can be solved by traditional algorithms. Since the complexity of the algorithm depends on the size of state space, we achieve savings in the computational as well as input cost compared to directly solving the original MDP. The model minimization methods concern with grouping behaviorally equivalent states only while leaving the original actions and observations unchanged. We are interested in the POMDP symmetry that is not related to reducing the size of the POMDP, nonetheless can be exploited to speed up conventional POMDP algorithms.

Model minimization techniques search for a homomorphism ϕ that maps M to another equivalent POMDP $M' = \langle S', A', Z', T', O', R' \rangle$ with the minimal model size. Formally, homomorphism ϕ is defined as $\langle f, g, h \rangle$ where $f : S \rightarrow S'$ is the function that maps the states, $g : A \rightarrow A'$ maps the actions, and $h : Z \rightarrow Z'$ maps the observations. Note that M' is a reduced model of M if any of the mappings is many-to-one. Depending on the definition of homomorphism ϕ , we obtain different definitions of the minimal model. Homomorphism ϕ defined only on the state spaces; the action and observation spaces remain the same as the

original POMDP. Hence ϕ takes a form of $\langle f, 1, 1 \rangle$ where 1 denotes the identity mapping. f should satisfy the following constraints in order to hold equivalence between M and M' :

$$T'(f(s), a, f(s')) = \sum_{s'' \in f^{-1}(s')} T(s, a, s'')$$

$$R'(f(s), a) = R(s, a)$$

$$O'(f(s), a, z) = O(s, a, z)$$

Table 1.2: Transition probabilities of the tiger problem

$T(s, a_{LISTEN}, s')$	$s' = s_{LEFT}$	$s' = s_{RIGHT}$
$s = s_{LEFT}$	1.0	0.0
$s = s_{RIGHT}$	0.0	1.0
$T(s, a_{LEFT}, s')$	$s' = s_{LEFT}$	$s' = s_{RIGHT}$
$s = s_{LEFT}$	0.5	0.5
$s = s_{RIGHT}$	0.5	0.5
$T(s, a_{RIGHT}, s')$	$s' = s_{LEFT}$	$s' = s_{RIGHT}$
$s = s_{LEFT}$	0.5	0.5
$s = s_{RIGHT}$	0.5	0.5

Table 1.3: Observation probabilities of the tiger problem

$O(s, a_{LISTEN}, z)$	$z = z_{LEFT}$	$z = z_{RIGHT}$
$s = s_{LEFT}$	0.85	0.15
$s = s_{RIGHT}$	0.15	0.85
$O(s, a_{LEFT}, z)$	$z = z_{LEFT}$	$z = z_{RIGHT}$
$s = s_{LEFT}$	0.5	0.5
$s = s_{RIGHT}$	0.5	0.5
$O(s, a_{RIGHT}, z)$	$z = z_{LEFT}$	$z = z_{RIGHT}$
$s = s_{LEFT}$	0.5	0.5
$s = s_{RIGHT}$	0.5	0.5

Table 1.4: Reward function of the tiger problem

$R(s, a)$	$a = a_{LISTEN}$	$a = a_{LEFT}$	$a = a_{RIGHT}$
$s = s_{LEFT}$	-1	-100	10
$s = s_{RIGHT}$	-1	10	-100

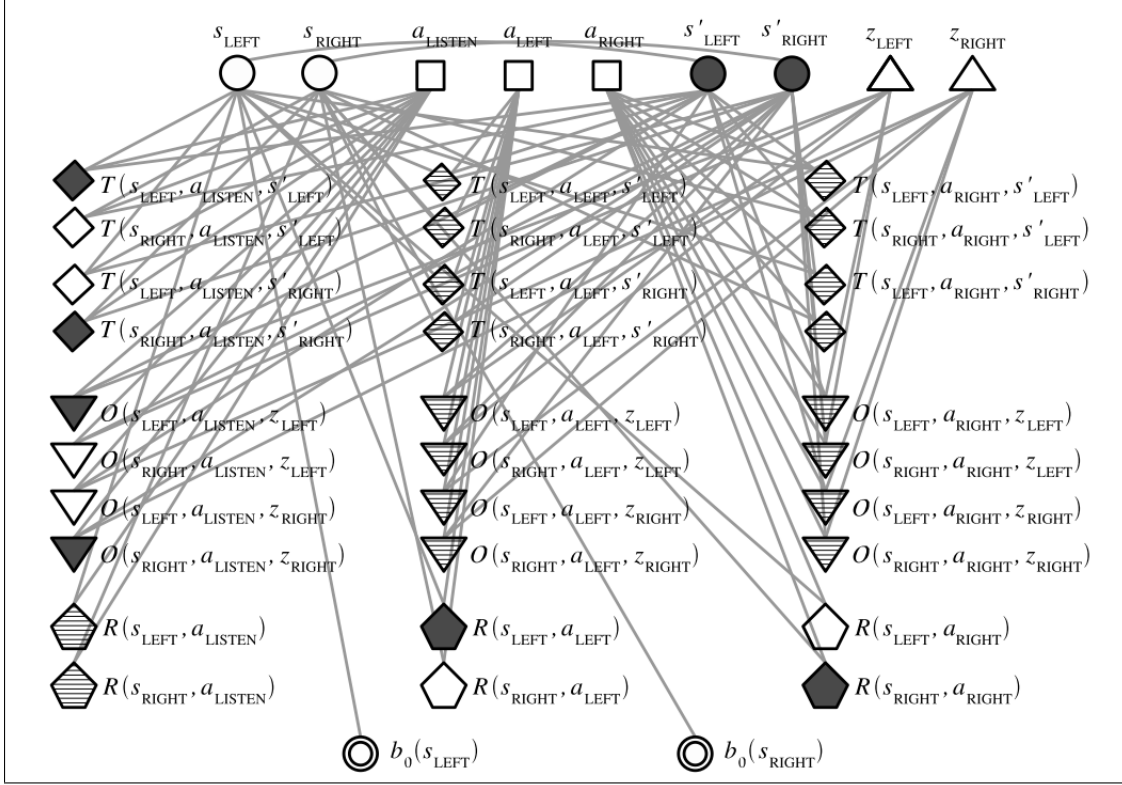


Figure 1.4: Encoding the tiger problem as a vertex-colored graph. Two vertices have the same color if and only if their shapes and fillings are the same.

An **automorphism** is a special class of homomorphism. In the context of POMDPs, an automorphism ϕ is defined as $\langle f, g, h \rangle$ where the state mapping $f : S \rightarrow S$, the action mapping $g : A \rightarrow A$, and the observation mapping $h : Z \rightarrow Z$ are all one-to-one mappings. Hence, ϕ maps the original MDP to itself, and there is no assumption regarding the reduction in the size of the model. The classic tiger domain [4] is perhaps one of the best examples to describe automorphisms in POMDPs. The state space S of the tiger domain is defined as s_{LEFT}, s_{RIGHT} , representing the state of the world when the tiger is behind the left door or the right door, respectively. The action space A is defined as $a_{LEFT}, a_{RIGHT}, a_{LISTEN}$, representing opening the left door, opening the right door, or listening, respectively.

The observation space Z is defined as z_{LEFT}, z_{RIGHT} representing hearing growl from the left or hearing growl from the right, respectively.

The specifications of transition probabilities, observation probabilities, and the rewards are as given in Figure 1.2, Figure 1.3 and Figure 3.1 respectively on page 11. The initial belief is given as $b_0(s_{LEFT}) = b_0(s_{RIGHT}) = 0.5$.

s_{LEFT} and s_{RIGHT} can be interchanged to yield an equivalent POMDP, simultaneously changing the corresponding actions and observations:

$$f(s) = \left\{ \begin{array}{ll} s_{RIGHT} & \text{if } s = s_{LEFT} \\ s_{LEFT} & \text{if } s = s_{RIGHT} \end{array} \right\}$$

$$g(a) = \left\{ \begin{array}{ll} a_{LISTEN} & \text{if } a = a_{LISTEN} \\ a_{RIGHT} & \text{if } a = a_{LEFT} \\ a_{LEFT} & \text{if } a = a_{RIGHT} \end{array} \right\}$$

$$h(z) = \left\{ \begin{array}{ll} z_{RIGHT} & \text{if } z = z_{LEFT} \\ z_{LEFT} & \text{if } z = z_{RIGHT} \end{array} \right\}$$

1.3 CONTRIBUTIONS

We shall attempt to make the following contributions in this thesis:

1. We propose a new algorithm based on Analytical hierarchical process to elicit user preferences of actions for every state and to map them to reward function with values generated by the algorithm.

2. We propose an approach of applying symmetry in POMDPs by designing a new algorithm for finding symmetries in the action space of the problem domain and suggesting them to the user in order to reduce the number of transitional probability inputs required by the user.
3. We present an approach to map user observations in the problem domain as a list of state preferences that the user believes to be observed on performing specific actions and then generating probability distribution over the state space for every action based on user beliefs about the observations.
4. We design Rae to have voice/text based decision-making support assistant using Rasa API stack that helps users as a decision-aid in converting their decision problem into a POMDP.
5. We present an intuitive way of interactive rational decision policy exploration technique to represent a circular graph as an infinite tree of action/observation choices.
6. We discuss hypotheses to find the requirements of a decision-aid systems and how our approach proves or disproves any of the hypotheses based on human usability study conducted under the Institutional Review Board at University of Georgia.

1.4 THESIS OUTLINE

Chapter 3 describes our decision-aid system called Rae, a rational decision making agent and our approach for eliciting human decision problems to POMDP. We explain our algorithm of preference elicitation to map state-action preferences to reward matrix.

We explain our approach of generating transition function from the user's input which we try to minimize using symmetry of actions and our approach of pivotal-transitions. We explain how we implemented our approach and how symmetry of actions could reduce the required number of inputs from the user for generating transition matrix of the problem domain.

We discuss our approach of eliciting user observations into an observation matrix. We describe our process of how we can generate observation matrix without asking for probability values of observing variables. We explain our algorithm and describe our approach.

We introduce our design of interactive policy and explain the approach. This interactive policy is where the users get to see the results of their decision problem and we try to evaluate our model based on policy satisfaction of the users.

Chapter 4 explains our experiment and results of analyzing and evaluating our model and approach of decision-aid for POMDPs by conducting a Human Study through Institutional Review Board (IRB) at University of Georgia.

Chapter 5 concludes our research and explains future work that can help improve our decision-aid model.

CHAPTER 2

RELATED WORK

2.1 INTRODUCTION

According to Phillips-Wren (2012) [17], Decision making is an inherently human activity that can have significant impacts. It is not surprising that researchers have attempted to improve the quality of decisions by developing computer technologies to augment and extend human capabilities. Advances in Artificial Intelligence (AI) have made this goal a reality in many applications. These AI-integrated decision making support systems, or intelligent decision support systems (IDSS), are increasingly used to assist decision making in areas such as finance, healthcare, marketing, commerce, command and control and cyber-security [17].

We discuss a few artificial decision making agents that are closely related and have impacted in designing of Rae which is our approach for designing an interactive decision-making agent to elicit any human decision problems into POMDPs.

2.2 ECONOMIC AGENTS

Arthur (1991) [18], explains a sample problem of iterated choices under uncertainty, in which a decision maker chooses one of N possible actions at each time that have random payoffs or profits drawn from stationary distribution that is unknown in advance. An artificial single-agent is designed to undertake single action out of N actions each time, learns by updating

the probabilities of taking each action on the basis of payoffs or outcomes experienced. It associates a vector of strengths S_i , with the actions 1 to N at each time t . The current sum of these strengths is C , (the component sum of S_t), and the initial strength vector S_0 is strictly positive. The vector p_t represents the agent's probabilities of taking actions 1 through N at time t . At each time t , the agent:

1. Calculates the probability vector as the relative strengths associated with each action. That is, it sets $p_t = \frac{S_t}{C_t}$
2. Chooses one action from the set according to the probabilities p_t and triggers that action.
3. Observes the payoff received and updates strengths by adding the chosen action's j 's payoff to action j 's strength. That is, where action j is chosen, it sets the strengths to $S_t + \beta_t$ where $\beta_t = \phi(j)e_j$ (e_j is the j th unit vector).
4. Re-normalizes the strengths to sum to a value from a pre-chosen time sequence. In this case, it re-normalizes strengths to sum to $C_t = Ct^v$.

Based on the above listed criteria, this paper concludes to design artificial learning agents and calibrate their rationality to replicate human behavior. Not only does the learning behavior of the calibrated agents varies as payoffs change from experiment to experiment in the repeated multi-choice context, but it also reproduces two stylized facts of human learning well-known to psychologists: that with frequency-dependent payoffs, humans meliorate rather than optimize; and there is a threshold in discrimination among payoffs below which humans may lock in to sub-optimal choices.

Criteria for Economic agents:

1. **Ease of use:** This paper describes an economic agent design to calibrate the agent's rationality that replicates human behavior. Hence, it can be said based on the design explained by Arthur (1991) [18] that this model is not easy to be used as the parameters are not easily configurable. This model contrasts to the elicitation technique that has been developed for Rae. Rae can be better in ease-of-use as compared to economic agent model.
2. **Accuracy:** According to the results discussed in this paper, it takes 40-100 more trials to get definitive results. Moreover, there is also a time horizon over which the economic environment of a decision problem stays relatively constant. Thus, the authors conclude that the agent's learning may take place more slowly than the rate at which problem shifts and could be less accurate. Comparing Rae to this model, we can say that this model is able to replicate human behavior satisfactorily while Rae doesn't recommended policy based on human behavior hence the acceptance of accuracy based on human behavior could be achieved better by the model discussed by Arthur (1991) [18] than our approach of Rae.
3. **Flexibility of agent (problem domain specificity):** The economic agent is designed specifically for a problem as discussed in the paper. It may be flexible in terms of parameters but the model cannot be modified as it tries to calibrate rationality based on human behavior. Rae is comparatively flexible in terms of domain but the policy suggested may not reflect human behavior as this agent's design.
4. **Helpfulness of agent:** This approach is helpful in designing an artificial decision making agent that could replicate human behavior in its policy. This can be helpful in a case when there are many choices that humans sometimes ignore and choose an inferior choice since the other choices were not know by them. In such a case, this agent gives results that could consider such inferior choices with a greater value in order to

choose actions based on human behavior. This could mean the agent’s recommendation could be more acceptable. Rae, on the other hand recommends rational policy based on the elicitation. Thus, the policy may or may not always be acceptable but is a satisfactory prescription.

5. **User preferences:** This agent doesn’t take into account the user preferences but tries to learn them through parameter calibration. This could be a good approach for modeling user preferences based on parameter calibration to replicate human behavior but given the model’s accuracy, it may need more configurations. Rae asks users for their preferences instead of approximating them since every user has their unique set of preferences and hence the policy recommendation differs to every user even though with they may have the same problem domain.

We observe that the agent design proposed by Arthur (1991) [18] tries to replicate the human behavior while Rae is designed to elicit any decision problem into a POMDP model. Also, the agent this paper proposes learns on choosing actions and learning outcomes while Rae provides a rational policy based on elicitation of problem domain of the user and hence policy recommends actions based on user’s observation, preferences of actions for each state and transition probabilities that user elicits based on their understanding of the domain problem.

2.3 PARK MANAGER AGENTS

Briot et al. (2009) [19] proposes an artificial agent architecture to implement ‘park manager’ cognitive decision rationale. This paper presents the *SimParc* project, a role-playing serious game aimed at computer-based support for participatory management of protected areas. The lack of human resources implied in RPG gaming process acts as a constraint to the fulfillment of pedagogical and epistemic objectives. In order to guarantee an effective learning

cycle, park manager role must be played by a domain expert. Required expertise obviously narrows game's autonomy and limits its context of application. Briot et al. (2009) [19] came up with a solution to this problem by inserting artificial agents into the game.

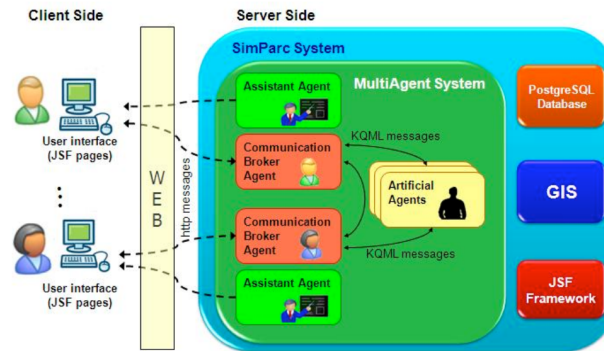


Figure 2.1: Sim Parc general Architecture by Briot et al. (2008) [19]

In the current architecture of the artificial park manager, only static information about the park and about the votes of the players are considered. Authors of the paper consider exploring how to introduce dynamics in the decision model, taking into account the dynamics of negotiation among the players (the evolution of player's decisions during negotiation). The artificial agent is an intelligent assistant agent designed to assist a player by performing two main tasks:

1. To help participants in playing the game, e.g.: the assistant agent tells the player when he should make decisions; what are the phases of the game; what should be done in each phase; etc.;
2. To help participants during the negotiations. For this second task, Briot et al. (2008) [19] would like to avoid intrusive support, which may interfere in the agent's decision making cognitive process and hence selected some objectives, e.g., to identify other players' roles with similar or dissimilar goals, which may help the human player to find possible coalitions or conflicts.

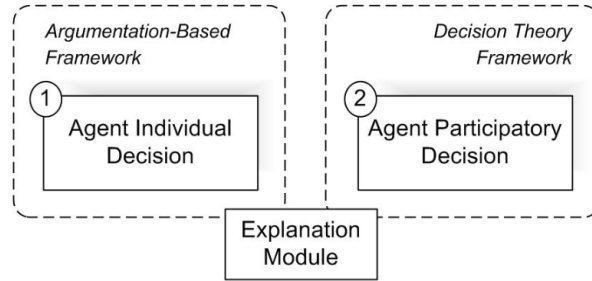


Figure 2.2: Park manager decision process by Briot et al. (2008) [19]

This paper presents the SimParc project which is a role-playing serious game aimed at computer-based support for participatory management of protected areas. Authors of this paper present a solution to insert artificial agents into the game and this paper focuses on architecture of implementing park manager cognitive decision model. Briot et al.(2009) [20] implemented the project and tested evaluation of their architecture. The game intends to be a tool capable of contributing to the dialogue on consolidation of commitments to conservation, particularly management of national parks and other protected areas. Results have shown that quick and simple solutions to modeling the complexity of this process can become a great risk of loss of meaning of the game.

Approach presented by Briot et al. (2009) [19, 20] is an architecture on designing a cognitive artificial agent for Park management. Our approach of designing Rae is different since Rae is not a cognitive model and is not domain specific. But the approach of designing a game and adding artificial agents could be a step further in building efficient cognitive model of Rae. Rae assists in elicitation of their problem domain and users need to come up with state, action, and observation labels on their own and also the probabilities of transitions are based on their judgment.

Introducing intelligent agents as discussed in this paper could help in better elicitation of the problem domain but our focus on designing artificial decision making agent Rae, is to make it more general with respect to the scope of decision problem and hence it would be difficult to design and introduce cognitive intelligent agents that are not domain specific.

Criteria for Park management agent:

1. **Ease of use:** As per the tests conducted as mentioned in this paper, the results were positive about the usability and ease of use of this model. Test was conducted with 10 people out of which 8 were domain experts and two were not experts. Out of those two, one had a very good idea of games while the other one was a complete beginner. Comparing to our agent Rae, this has a different setup which is based on a game setup where players enter their parameters in a simulation of Park management.
2. **Flexibility of agent (problem domain specificity):** This approach is open to be used for other problem domains but it needs to be configured for other domains and it cannot be easily adopted for other domains. We can say the approach authors have mentioned in this paper is vital in designing artificial intelligent agents in a game setup. Configuring those agents is a task this approach requires to adapt those agents in a new domain.
3. **Helpfulness of agent:** After evaluation of the tests, authors of this paper conclude that more information and interaction was required by participants while playing the park management game setup with intelligent agents. The biggest difficulty encountered by the participants was the understanding of different zoning and hence it was highlighted the importance of improving access to information for each player.

4. **User preferences:** SimParc model requires a continuous and dynamic interaction between players and it has been highlighted the importance of the use of flexible systems with additional design features which the participants believe would enhance the process of negotiation in the game.

2.4 ARTIFICIAL DECISION MAKING FOR INTELLIGENT BUILDINGS

Boman et al. (1999) [21] design an automated decision support system for intelligent buildings to increase the energy savings and to improve the customer value-how the people in the building experience the effects of actions of the agents. A number of assumptions related to agent control, through monitoring and delegation of tasks to other kinds of agents, of rooms at a test site are relaxed. Each assumption controls at least one uncertainty that complicates considerably the procedures for selecting actions part of each such agent.

Authors of this paper use a multi-agent system approach to intelligent building control. The multi-agent approach allows for a structure-preserving mapping of the design entities of the application and of the smart equipment of the implementation. It is an open architecture in which agents can be easily configured and re-configured, even dynamically. It is also truly distributed, since the authors make no assumptions about the locations of the agents. Usually, the goal of a Room agent and agents realizing user preferences in the room are conflicting: The Room agent tries to maximize energy savings while other agents try to maximize customer value. In the intelligent building domain, this is the main trade-off. Then *preferences* of each person in the building are encoded in a *Personal Comfort agent*. The agents make calls to a pronouncer which provides extremely fast decision support by evaluating the input (a decision tree or influence diagram) and returning the best actions. Borman et al. (1999) [21] investigated a number of commercial academic tools to provide interfaces suitable for agent interaction and test run their agent on three tools- Netica, SMILE and DATA Interactive.

Netica and SMILE are based for finding optimal policy in an influence diagram. DATA Interactive uses decision trees to represent decision problems, and is based on averaging out and folding back algorithm.

Rae provides the recommended policy in an interactive way of infinite tree which is a better representation than Netica, SMILE or DATA Interactive softwares. Rae, provides an interactive way of exploring the policy based on their observations. Rae has a benefit where the parameters are elicited by the user and hence, Rae is not domain-specific for solving decision problems.

Criteria for Intelligent Building agent:

1. **Speed:** Approach presented by this paper is faster even on the slowest tool- SMILE which solves the decision problem at hand in less than ten milliseconds. With this short response time, decision agents could make extensive use of decision support provided by the pronouncer without any noticeable degradation of the system. This paper demonstrates that the use of pronouncers is feasible in many domains and to implement them as well. Comparing speed to Rae, this approach is indeed faster than Rae. But Rae's response of the recommended policy doesn't take longer than few seconds and depends on the problem domain. More the number of states, actions or observations, more the time is required by Rae but it won't take any longer than few seconds.
2. **Accuracy:** This paper proposes a multi-agent approach that would improve efficiency of power usage in an intelligent building and the agent designed makes use of user preferences and provides assistance. This is very similar to how Rae recommends a rational policy. Rae, takes into consideration the user preferences of actions for each state which are mapped to reward values and assists users in eliciting their decision problem into a POMDP model.

3. **Flexibility of agent (problem domain specificity):** Approach of this paper is not domain-specific, and authors of this paper designed the agent for improving energy efficiency and maximizing customer value of intelligent building but it can be applied for different domains as suggested by the authors.

2.5 AI FRAMEWORK FOR CLINICAL DECISION-MAKING

According to Bennet et al. (2013) [22], the modern healthcare systems are rapidly expanding costs/complexities, growing myriad treatment options and exploding information streams that often do not effectively reach the front lines hinder the ability to choose optimal treatment decisions over time. This paper develops a general purpose (non-disease-specific) computational/artificial intelligence (AI) framework to address these challenges. The approach combines Markov decision processes and dynamic decision networks to learn from clinical data and develop complex plans via simulation of alternative sequential decision paths while capturing the sometimes conflicting interactions of various components in the healthcare system. It can operate in partially observable environments (in case of missing observation or data) by maintaining belief states about patient health status and functions as an online agent that plans and re-plans as actions are performed and new observations are obtained.

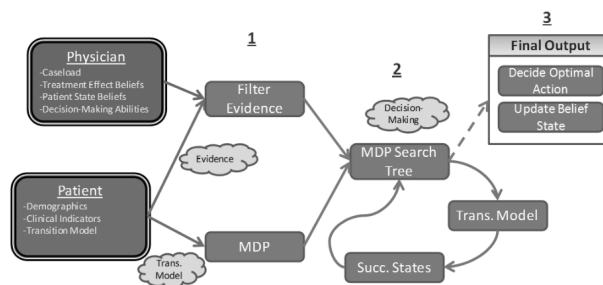


Figure 2.3: Architecture of MDP for clinical decision making [22]

The algorithm presented by authors Bennet et al. (2013) [22] is as follows:

1. Create patient and physician agents
2. Create patient-specific MDP.
Then, for each timepoint (if not horizon):
3. Calculate current outcome delta, physician agent filters evidence
4. Determine optimal current action via MDP search tree
5. Perform action and update belief states
6. If action \neq not treat, return to step 3

The POMDP approach modeled for the decision-making environment in this paper [22] is finite-horizon, undiscounted, sequential decision-making process in which the state s_t from the *state space* S consists of patient's health status at time t . At each time step the physician agent makes a decision to treat or stop treatment (an action a_t from the binary action space $A = 0, 1$). Here time corresponds to the number of treatment sessions since the patient's first visit. The physician agent receives rewards/utilities which are *Cost per unit change (CPUC)* which measures the cost in dollars it takes to obtain one unit of outcome change (*delta*) on a given outcome, and is asked to pick actions in order to maximize overall utilities. It is modeled as a dynamic decision network (DDN). The results of this approach demonstrate the feasibility of such an approach relative to human decision-making performance. An AI framework easily outperforms the current treatment-as-usual case-rate/fee-for-service models of healthcare [22].

This paper provides a standard approach in clinical decision making using POMDPs and explains the architecture of parameters that relate to the healthcare systems. This paper motivates in designing Rae to be non-domain-specific. Transitions and rewards were calculated based on static information of clinical systems and hence the authors of this paper Bennet et al. (2013) [22] are interested in having personalized transition models, variable

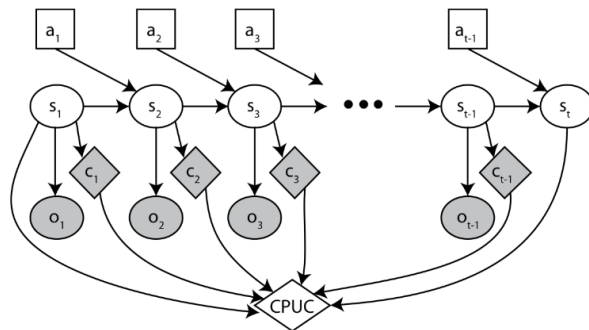


Figure 2.4: A dynamic decision network for clinical decision-making [22].

physician agents, improved non-deterministic choices, better state conceptualization and better utility conceptualization. Our research in designing Rae has been on effective elicitation of states, actions, transitions, rewards/utilities, observations and policy of the decision problem. Our research may help in improving clinical decision-making by using Rae to elicit clinical decision problem into a POMDP which will have all the possible factors as listed in this paper along with author's interests.

Criteria for clinical decision-making aid:

1. **Feasibility:** The results of this approach demonstrate the feasibility where such an AI framework easily outperformed the current treatment-as-usual (TAU) case-rate/fee-for-service models of healthcare where CPUC was \$189 vs. \$497 for AI vs. TAU where lower is considered optimal.
2. **Accuracy:** This approach was able to obtain 35% increase in patient outcomes. Tweaking certain AI model parameters could further enhance this advantage, obtaining approximately 50% more improvement for roughly half the cost.

3. **Flexibility of agent (problem domain specificity):** Given the multi-agent design, this system can be modeled to an individual, personalized treatment basis (including genetics), i.e. *personalized medicine*. But this model is restricted to clinical domain but we can still say that this model is flexible in terms domain problems as we only need to tweak a specific parameters to adopt for another decision-making aid in clinical domain space.

2.6 DXPLAIN: DIAGNOSTIC DECISION-SUPPORT SYSTEM

DXplain is an evolving computer system that uses an extensive knowledge base and relatively simple computational models to provide significant diagnostic problem-solving assistance to the practicing physician [23]. According to authors of this paper Barnett et al. (1987) [23], the program allows the computer-naive user to enter a set of patient signs and symptoms and then generates a list of hypotheses that deserve consideration. This system comments on the diagnostic relevance of each sign and symptom and suggests specific additional data elements that might clarify differential diagnosis currently favored by the computer model.

DXplain is a web-based, diagnostic decision support system used at Massachusetts General Hospital/Harvard Medical School. As a decision support tool, DXplain uses Bayesian logic and its knowledge base of the crude probabilities of over 5,000 clinical manifestations associated with over 2,000 different diseases. These data describe the relationships between symptoms, signs and lab findings, and the diseases which they are a part.

One of the most commonly used functions is *Case Analysis*. In this mode, the program produces a ranked list of diagnoses which might explain (or be associated with) the clinical findings entered by the user. DXplain uses an interactive format to collect clinical information and produces a ranked list of diagnoses which might be associated with the clinical manifestations. It also provides justification for each disease.

Criteria for DXplain:

1. **Ease of use:** DXplain is a web based interface designed for computer-naive users as a decision support tool for clinical findings. It has a very simple interface that states the requirement of signs and symptoms to be entered by the user. This system is recommended to be used only by physicians who can enter correct signs or symptoms. Comparing this model with Rae, we can say that DXplain provides better ease-of-use since it has not too many inputs required for the diagnoses and it is expected that the user is a physician who is an expert of the clinical diagnoses and can accurately elicit signs and symptoms of their patient into the required format by DXplain decision support tool. Rae, on the other hand doesn't require user to be an expert of the domain and hence has more parameters and elicitation requirement than DXplain.
2. **Accuracy:** DXplain is used as an assistive tool by physicians to understand different hypotheses of signs and symptoms to rule-in or rule-out different hypotheses and to confirm the findings. Hence, this model is not offer definitive medical consultation and is not recommended to be used as a substitute for physician diagnostic decision-making. Its accuracy is largely dependent on the knowledge database and diagnostics. It is continually being improved and adapted as a result of comments from the users. Accuracy of DXplain depends on its knowledge base which is continuously improved based on the user's comments. Hence, we can conclude that this model provides accurate diagnoses based on physician's ability of entering accurate signs and symptoms into the model. Contrasting with Rae, we observe that Rae doesn't have a knowledge

base of the problem domain and hence its accuracy can be evaluated on policy satisfaction of the user. It also depends on how well the user receives the recommended policy since the policy is completely based on user's accuracy of domain elicitation.

3. **Flexibility of agent (problem domain specificity):** This model is flexible in the domain space of clinical diagnoses. This tool can be used for ranked hypotheses of diagnoses based on evaluation of signs and symptoms entered by the physician. Hence, this tool helps in diagnoses of many diseases in clinical domain. This model although flexible, is still restricted to clinical diagnoses of diseases and this model can't be used for any other purposes.

In contrast with Rae, this model is domain-specific and is used as an assistive tool in disease diagnoses by physicians who can interpret diagnoses based on signs and symptoms. This assistive decision-making tool thus helps user in improving confidence over a specific diagnoses or suggesting different diagnoses based on signs and symptoms that may be considered based on the provided ranks. Rae is not domain specific and can be used for any problem domain. This may give less satisfying policy recommendations since Rae, even though domain specific, has no knowledge base of the problem domain but it can provide satisfactory policy recommendations to the user for any problem domain.

4. **Helpfulness of agent:** DXplain has been proved to provide assistance in disease diagnoses to physicians and given its huge knowledge base (KB of about 200,000 individual data points in the KB representing disease/finding relationships, it helps physicians in crude approximation of disease prevalence and other important features.

DXplain is helpful for physicians in ruling in and out a few diagnoses based on signs and symptoms of the patients. Knowledge base is continuously improved based on physician's comments and results.

Rae can help in assisting users for eliciting their problem domains into a POMDP model and we explain our results based on human usability study that we conducted. Rae may not provide different diagnoses like DXplain but can assist in providing

5. **User preferences:** User preferences have significant importance in DXplain tool for diagnoses since the physicians who enter information, provide their view and preference of signs and symptoms based on which this model provides the diagnoses assistance. This is very similar to Rae as user needs to enter their preferences of actions for each state. Thus, DXplain and our model, both take user preferences into account.

2.7 ACS NSQIP SURGICAL RISK CALCULATOR

The ACS NSQIP surgical risk calculator is a decision-support tool based on reliable multi-institutional clinical data, which can be used to estimate the risk of most operations. It allows clinicians and patients to make decisions using empirically derived, patient-specific postoperative risks [24]. According to the authors Bilimoria et al. (2013) [24], accurately estimating surgical risks is critical for shared decision making and informed consent.

Using standardized clinical data from 393 ACS NSQIP hospitals, a web-based tool is proposed in this paper that allows surgeons to easily enter 21 preoperative factors such as demographics, comorbidities, procedure, etc. Regression models are developed to predict 8 outcomes based on the preoperative risk factors. Authors Bilimoria et al. (2013) [24] used *Random intercept, fixed slope hierarchical models* (using SAS GLIMMIX), which account for clustering of cases within hospitals and impose an empirical Bayes-type shrinkage adjustment. Only fixed (patient's level) effects were used for risk prediction, although the methodology would permit inclusion of hospital-specific effects as well.

According to Bilimoria et al. (2013) [24], because the postoperative complications risks estimated by the surgical risk calculator may not capture every potential comorbidity, the authors of this paper sought to create an ad-hoc opportunity for surgeons to reasonably modify the estimated risks. Given that there is a degree of uncertainty (confidence level) around the estimated risks, authors allowed clinicians to increase the risk of surgery within the confidence interval for each specific CPT. The default estimated risk from the model is designated as a surgeon adjustment score (SAS) of 1. The surgeon can then increase the risk to an SAS of 2 (+1 standard deviation of predicted risks for that CPT) or to an SAS of 3 (+2 standard deviation).

The ACS NSQIP surgical risk calculator offers surgeons the ability to quickly and easily estimate important, patient-specific postoperative risks and present the information in a patient-friendly format. Discussions of these risks may better inform patient and caregiver expectations, help surgeons and patients decide which operation to perform, and even offer insights about whether the operative risk is prohibitive. The surgical risk calculator offers an opportunity to improve shared decision making and informed consent [24]. Comparing this to Rae, we observe that this model is a shared decision making risk assessment model which requires a surgeon's risk assessment to increase or decrease the risk assessment value suggested by the model while Rae is completely dependent on the user's problem domain elicitation and by changing parameters, user can modify the recommended policy but this may not quantitatively hold true since the user may not be the domain expert of the decision problem and hence their judgment of parameters could be biased.

Criteria for risk calculator:

1. **Ease of use:** Model developed in this paper is easy to be used by clinicians and surgeons as the preoperative parameters are well defined and the postoperative risk assessment is presented with surgeon's validation. SAS feature of this model offers the surgeon an option to better counsel patients through shared decision making risk analysis.
2. **Accuracy:** Model provides good results but as SAS is a modification that is performed to augment the estimated risks outside of the modeling process, there is no quantitative evidence that these adjusted risks are more accurate.
3. **Flexibility of agent (problem domain specificity):** This model is domain specific and is developed specifically to assess postoperative risks involved and can be used efficiently and accurately by the surgeons. Although this model allows to change parameters, the domain of risk calculator cannot be changed since the parameters are designed for surgeons to enter their preoperative parameters along with their risk assessment. Comparing to Rae, this model will give better risk analysis but is much restricted to medical-surgery domain. Rae on the other hand may provide better assistance to patients in risk analysis based on their perspectives and may also get better policy if the model can be used by patients along with surgeons in correctly assessing their preoperative parameters in Rae to see a POMDP policy recommendation of risk assessment.
4. **Helpfulness of agent:** This approach helps surgeons, patients and patient's caretakers in assessing postoperative risks and needs that can be well discussed in preoperative interactions. Although the authors of this paper are working on improving patient's interaction with this model, to inform them of the risk assessment in a way that any patient can understand it in a better way and need not be a medical expert of the field. This model is definitely helpful given the results and shared decision making improves

risk assessment and helps surgeons in understanding postoperative risks involved along with their expectations. We designed Rae in a way in which users can interact with the policy to explore all the possible recommendations in different scenarios. Our approach of policy recommendation may help in improving this system for better assistance.

5. **User preferences:** This is a shared decision model and hence surgeon's preferences of risk assessment reflect in the postoperative risk analysis provided by this system. Our model of Rae does elicit user preferences but these preferences carry bias based on their experience and domain knowledge which may or may not be accurate as that of surgeons in the model presented by this paper.

2.8 DECISION AIDS FOR PATIENTS

According to authors Knops et al. (2013) [25], when consenting to treatment, few patients adequately understand their treatment options. To help patients make deliberate treatment choices, decision aids provide evidence-based information on the disease, treatment options, and their associated benefits and harms. Although decision aids are not designed to direct patients toward a particular treatment option, it is possible that their introduction will change the proportion of patients that opt. Offering a decision aid increases the number of patients who prefer conservative or less invasive treatment options and as the decision-aids improve patient's knowledge and lower decision conflict without without raising anxiety levels. This decision aid is important according to authors Knops et al. (2013) because in daily practice, surgeons often only have limited time to discuss complex matters of surgeries properly with patients. Information is often concentrated on the proposed treatment option, whereas alternative treatment options are not always mentioned and hence consenting to treatment, only a few number of patients adequately understand their treatment options and the associated risks and benefits.

Authors Ubbink, Knops et al. (2008) [26] discuss their approach of designing and developing an evidence-based decision aid to enhance shared decision making by patients with an asymptomatic abdominal aortic aneurysm and to inform them about the pros and cons of their treatment options. Starting April 2007, the pilot version was designed as a PowerPoint presentation (Microsoft Office 2003; Microsoft Corp., Redmond, WA). This comprised a total of 170 pages ('slides') and offers general information as to the disorder, the possible treatment options, and the opportunity to balance the risks involved [26].

According to this paper, the approach was to design and develop a decision-aid that could be very simple to be used, informative about the patient's clinical situation and proposing pros, cons and different options of treatment to the patient in a way to help them understand rather than to be afraid of the treatment. Results of this paper have been good where most of the patients are satisfied with the provided decision-aid for treatments and has helped them to understand the complexity or risks associated with their decisions. Goal of this paper was to design the interface as simple as possible where even the old-aged patients could use the digital system easily and this decision aid would help them in making shared decisions.

Criteria for decision-aid for patients:

1. **Ease of use:** Approach presented in this paper has a very user-friendly interface and focuses on making sure the large number of slides (170) of the power-point presentation don't drift off the patient from their decision aid and results have shown to prove their approach with good reception from the participants. Comparing with Rae, this interface design is very easy to use but this approach considers no input from the user but to only make decision choices that would lead to different information regarding the treatment, options, risks, pros or cons. The initial version of this decision support had limitations as to the complexity of the individual risk information, which was dissatisfying for the

patient. After adjustments, appraisal by AAA patients showed that this decision aid was considered user- friendly, of additional value, and improved understanding [26]. Rae, on the other hand elicits user decision problem and hence it requires users to understand the approach and to enter parameters to Rae for decision-making.

2. **Flexibility of agent (problem domain specificity):** This approach is flexible in terms of domain but it restricts the design by specifying alternatives which may not be modifiable by the participants. Thus this approach, although user friendly, is limited to the information and decisions designed by the expert designer of the decision-aid and the patients or user has less control over the decision-aid. Comparing this to Rae, we observe that the policy recommended by Rae is completely controlled by user based on his understanding of the problem. But, if the user has not completely understood the problem or the problem is not well defined then the policy recommended by Rae may be received satisfactorily.
3. **Helpfulness of agent:** Results have shown that this approach of shared decision-making for treatment options has helped vast majority of the patients as they stated it offered additional value in their decision making. In contrast with Rae, this approach was well received as the interface was very user friendly and the patient need not know about the decision problem completely and was more informed about the possibilities based on their treatment option and hence our approach of designing Rae may not be as helpful as this approach but both of these approaches present different scenarios where both have equal pros and cons.
4. **User preferences:** This approach does take user/patient preferences into account as designed by the expert of decision aid based on the available information but the individual user preferences are not considered. In only one study the benefits of comprehensive, highly individualized, evidence-based brochure for asymptomatic AAA patients

based on information by way of Markov analysis have been studied. The highly individualized information was found to be dissatisfying and impractical according to the authors Ubbink, Knops et al. (2008) [26]. Therefore, the information offered in this DA was mainly on headlines rather than focusing only on the individual situation of each patient. Comparing with Rae, our approach focuses on individual situations and considers user preferences to generate reward values of the POMDP. Thus, every individual has different rewards for their actions in specific state based on their understanding. But, our approach restricts the user-preferences based only on user's understanding which may also be biased and may not reflect the correct preferences the way this paper presents their approach where the preferences are information available in the treatment boundary.

CHAPTER 3

RATIONAL DECISION MAKING AGENT

In this chapter, we explain Rae, a rational decision making agent that we designed to elicit any complex decision making problem of humans into a POMDP model. We explain the concepts and approach of Rae beginning from states, actions, reward matrix, transitions, observations and policy. We show our model as viewed by the user and explain the concepts with an example domain of Tiger problem as explained on page 41.

3.1 VOICE/TEXT AGENT

Our model begins by introducing our agent Rae, who explains how to interact and proceed with the model. We designed Rae to have an interactive window that answers to users' queries through natural language processing and clarifies concepts, provides examples or responds to questions that users may have while interacting with the model. Rae supports speech-to-text input wherein the users can ask question in natural language through microphone and Rae converts their speech to textual question with the help of webkit-speech recognition JavaScript API developed by Mozilla in collaboration with W3 Standard [13]. This is still an experimental feature in our model.

Users can also ask questions by entering textual question in natural language in the Rae's interactive window. Rae has been trained to recognize and process natural language in context with POMDP model. We used Rasa API which is an open source Natural language processing and intent learning API developed in Python programming language. We develop

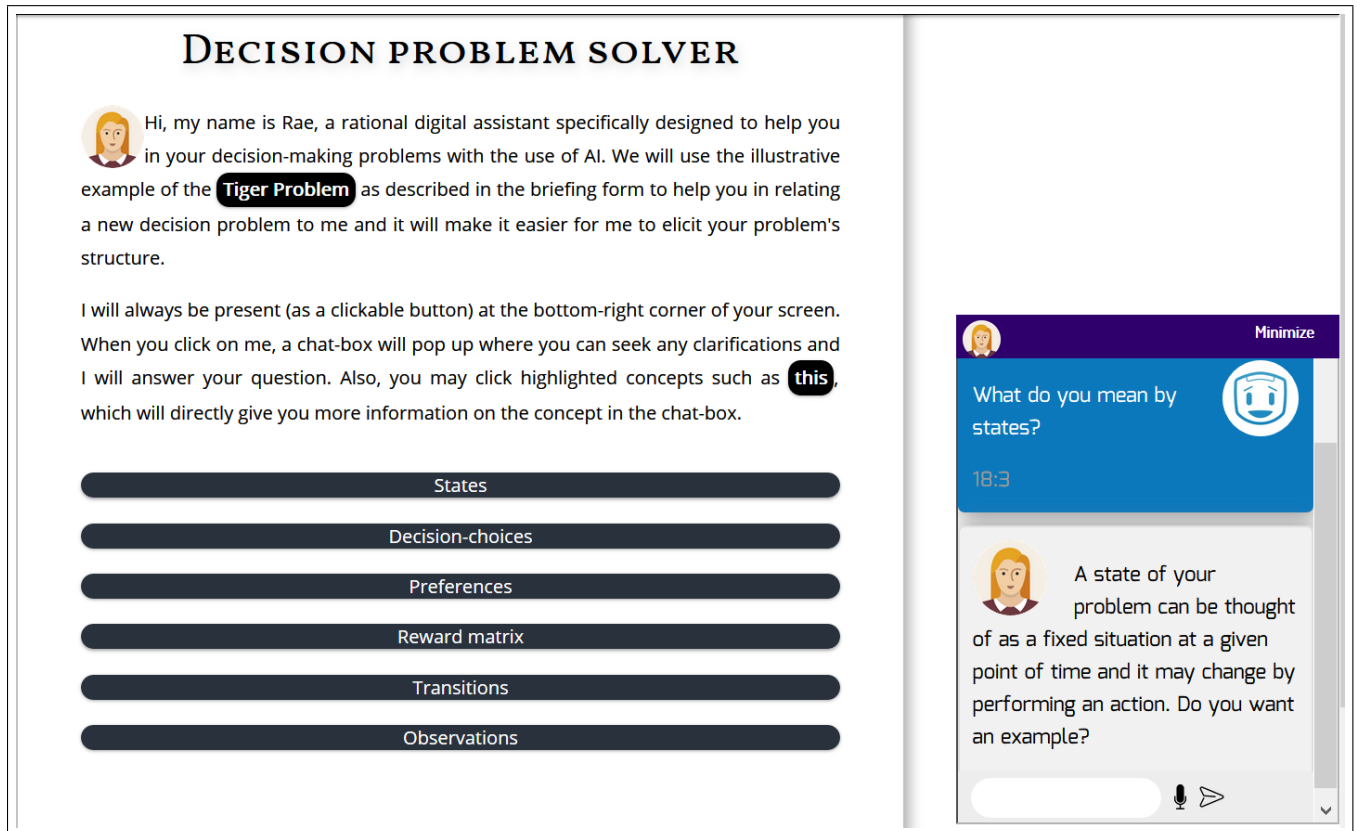


Figure 3.1: Model of Rae, as seen by the user before exploring any of the parameters

a model based on Rasa API [12] that learns and responds to the questions asked by user with reference to the model. Thus, Rae can learn, respond and interact with the user and provide clarifications to whatever questions the user may ask with respect to the model.

Our model is designed to answer questions with examples related to Tiger problem domain as explained on page 41. So the users can relate their complex decision problem with tiger problem and hence they can elicit with parameters such as states, actions and observations. A state of decision problem can be thought of as a fixed situation at a given point of time and it may change by performing an action. Users enter state labels of their choice that they

believe are the states of their problem. Then users proceed to enter action labels or labeled decision choices. Now, Rae knows about states and actions of the problem domain and hence we have states and actions of the POMDP. Now, we explain how we calculate reward matrix, transition matrix and observation matrix in the following sections.

3.2 PREFERENCE ELICITATION

Preference elicitation (PE) is a fundamental problem in the development of intelligent decision tools and autonomous agents. Software and agents of this type are often charged with the task of making decisions, or recommending courses of action, for a specific user. Making optimal decisions on behalf of a user requires knowing some information about her preferences or utility function. It is important to keep in mind that utility functions can vary widely from user to user (even while the other ingredients of a decision scenario, such as system dynamics, remain fixed across users). For this reason, preference elicitation—the process of extracting the necessary preference or utility information from a user is arguably one of the more important problems facing AI [6].

The image shows a user interface for the Rae model, divided into two sections. The top section is titled "States" and contains the text: "Enter labels for the **states** of your problem separated by commas in the following input box." Below this, there is an input field with the text "States: TL,TR" and a green "Done" button. A feedback message below the input field reads: "Good! Let's move to the next steps." The bottom section is titled "Decision-choices" and contains the text: "Enter the **decision choices** in your problem in the following input box." Below this, there is an input field with the text "Decision-choices: OL,OR,L" and a green "Done" button. A feedback message below the input field reads: "Good! Let's move to the next steps."

Figure 3.2: Model of Rae, after entering state and action labels into the model.

3.2.1 PREFERENCES TO REWARD FUNCTION

As an illustrative example consider the Tiger problem domain: Imagine you are standing in front of two closed doors. Behind one of the doors is a tiger and behind the other is a large reward. If you open the door with the tiger, then a large penalty is received (presumably in the form of some amount of bodily injury). Instead of opening one of the two doors, you can listen, in order to gain some information about the location of the tiger. Unfortunately, listening is not free; in addition, it is also not entirely accurate. There is a chance that you will hear a tiger behind the left-hand door when the tiger is really behind the right-hand door, and vice versa. For this example, we have states: Tiger behind the left door (TL) or Tiger behind the right door (TR), actions: Open left door (OL), Open right door (OR), Listen to growls (L), observations: Growl from the left (GL) or Growl from the right (GR).

Preferences

Did you finish entering your states and decision choices? You may go back up and make any changes you wish - be sure to click 'done'. Now, I will solicit your **preferences** in the decision problem. If there is no text box visible below then you haven't entered your states or actions. Enter your preferred actions below for each state.

State TL:

State TR:

Figure 3.3: Model of Rae, after entering state and action labels into the model, it asks users to enter preferences of actions to each state.

To understand preference elicitation, let's try to figure out preferences of actions for each of the states in the Tiger problem domain. For state 'TL', we would prefer to 'Open the right door (OR)' since we know the tiger is behind the left door or we could choose to 'Listen to growls (L)' if not confident that the Tiger is on the left but we would definitely not choose to open the left door in this state.

So, our order of preferred actions for state: TL is “OR, L” which means action OR is highly preferred over action L and action OL is not at all preferred to be performed in the state TL. Similarly, we observe that preferences for action TR would be OL, L.

Did you finish entering your states and decision choices? You may go back up and make any changes you wish - be sure to click 'done'. Now, I will solicit your **preferences** in the decision problem. If there is no text box visible below then you haven't entered your states or actions. Enter your preferred actions below for each state.

Good! Let's move to the next steps.

State TL:

State TR:

Figure 3.4: Model of Rae, after entering preferred actions for each state into the model.

Now, that we understood what preferences are and we found our preferences of actions for each state in the tiger problem, let's try to find the reward function for the tiger problem given our preferences of actions for each state. This is called 'preference elicitation' which will assign unique reward values for each state-action pair in the reward function.

According to our algorithm, we have states, actions and preferences provided by the user for the problem domain. Following is our preliminary reward matrix:

State \ Action	OL	OR	L
TL			
TR			

Let's calculate reward values for state **Tiger on the left (TL)**: Preference of actions for state **TL: Open right door (OR), Listen (L)** If any of the action from actions of the problem domain is not in the preferred actions provided by the user then it gets a negative value of the total size of *actions* + 1.

Actions that are preferred for that state are then compared with each other based on their '*index*' values, which means lower the index of that preferred action then it gets higher value than the action with higher index:

OR	L	
0	1	2

We observe that action 'OR' has index '0' while action 'L' has index '1'. Thus, action 'OR' is preferred over action L, and while OL is missing, it gets the lowest value as discussed above. Thus, as per our algorithm, we get the following matrix:

1	-4	-4
$\frac{-1}{4}$	1	4
$\frac{-1}{4}$	$\frac{1}{4}$	1

After we get the above matrix, we need to normalize it to generate a single list of consistency index values for our action set for state TL. We represent this list as 'geometricmean' list. Although the list does not contain geometric means, it reflects the AHP methodology of calculating geometric means for each row of the matrix which will be the λ_{max} values for that list. But, we have modified the process with our algorithm to find normalization instead of geometric mean since the values of geometric mean need to follow the C.I rule as discussed in the AHP process and hence we lose the information if the geometric mean is less than the minimum C.I value of the λ_{max} function. Thus, with our method, we calculate values without eliminating any of the values thus keeping it consistent with the C.I as well. We get the following matrix after normalization:

-2.52	1	0.39685
-------	---	---------

This is our translation of the above 2D matrix into 1D array. These are not the final values yet. Now we need to calculate the exact reward value for each action but from the above numbers we can get a rough image of what our reward function is going to look like for State TL.

-2.52	1	0.39685
OL	OR	L

We observe from the above partial array that action OL is having lowest reward, action OR is having highest reward and action L having better reward than action *OL* but less than action *OR*. We calculate reward values which are the ratio of $\frac{\text{geometricmean}}{\text{sum of geometricmeans}}$. Thus we get,

-2.24368	0.890353	0.3534787
----------	----------	-----------

We then multiply the above values with ‘100’ just to make them consistent with POMDP file structure and also to be readable for POMDP theory. Thus, we get the reward values for State TL with action preferences: OR, L as :

OL	OR	L
-224.39	89.05	35.34

Similarly, we get reward values for state *TR* with action preferences : *OL, L*as:

OL	OR	L
89.05	-224.39	35.34

We can represent the reward function as:

Table 3.1: Reward Function of Tiger Problem

State \ Action	OL	OR	L
TL	-224.39	89.05	35.34
TR	89.05	-224.39	35.34

Algorithm 1 Algorithm for generating Reward Function

```

1: procedure PROCESSREQUEST(request, response)           ▷ Get data as JSON array
2:   List < String >: states, actions
3:   Map < String, List < String >: preferences
4:   for each state in states do
5:     List : jarray ← Listofpreferencesforthatstate
6:     List : state_pref ← preferencesofactionsforthatstate
7:     preferences ← state_pref                       ▷ Add to preferences map
8:   States: state, Actions: action, PreferenceElicitation: pf   ▷ Create objects of class
9:   procedure PREFERENCEELICITATION(state, action, pf) ▷ Initialize preference map
10:  Matrix(state, action, pd): m                               ▷ [Create object m of class Matrix]
11:  m.generateMatrix()
12:  procedure GENERATEMATRIX
13:    for each state in states do
14:      for  $i = 0, i < actions.size()i ++$  do
15:        matrix.put(i, new List)
16:        for  $j = 0, j < actions.size(), j ++$  do
17:          if i equals j then
18:            matrix.add[state, j, 1.0]
19:          else
20:            if preferredActions doesn't contain i then
21:               $d \leftarrow -1 \times (actions.size + 1)$ 
22:            if preferredActions doesn't contain j then
23:               $d \leftarrow -1 \times \frac{1}{(actions.size+1)}$ 
24:            if actions.indexOf(i) < actions.indexOf(j) then
25:               $d \leftarrow (actions.size + 1)$ 
26:            else
27:               $d \leftarrow \frac{1}{(actions.size+1)}$ 
28:            Matrix ← state, action, preferences, d
29:  Calculator(matrix): cal                                     ▷ Create object cal of class Calculator
30:  for i=0, i< matrix.size(), i++ do calc(matrix.getMatrix.get(list))
31:  procedure CALC(list)
32:    sum ← 0.0, mul ← 1.0, root ← 0.0
33:    for i=0, i< list.size(), i++ do
34:      sum ← sum + list.get(i)
35:      mul ← mul × list.get(i)
36:    if mul < 0 then
37:      mul ← mul × -1
38:      root ←  $mul^{\frac{1}{list.size()}}$ 
39:    if sum < 0 then
40:      root ← root × -1
41:    geometricMean ← root
42:  result()

```

```

43: procedure RESULT
44:    $sum \leftarrow 0$ 
45:   for element in geometricMean do
46:      $sum \leftarrow sum + geometricMean(i)$ 
47:   if  $sum < 0$  then
48:      $sum \leftarrow sum \times -1$ 
49:   for every element in geometric mean do
50:      $ratio \leftarrow \frac{geometricMean(i)}{sum}$ 
51:      $geometricMean \leftarrow (i, ratio)$ 
52:   return  $geometricMean$ 

```

Algorithm 1 proceeds when the user completes entering preferences of actions for each state, the states, actions and preferences of these actions are sent to the function *processRequest* on line 9 of the algorithm. It creates a matrix that would store values of rewards for states and actions. Function *generateMatrix* creates matrices for every action. This matrix defines the value for that particular action. Then, we use our approach to calculate a value of the whole matrix based on user preferences for each state. These are stored in lists for every action. These lists are sent to the function on line 31 called *Calc* which then associates value after calculation for every action to every state. Function *result* on line 43 then generates a value for every state action pair which would be the reward value for choosing to perform an action in a particular state.

3.2.2 ARCHITECTURE OF PREFERENCE ELICITATION

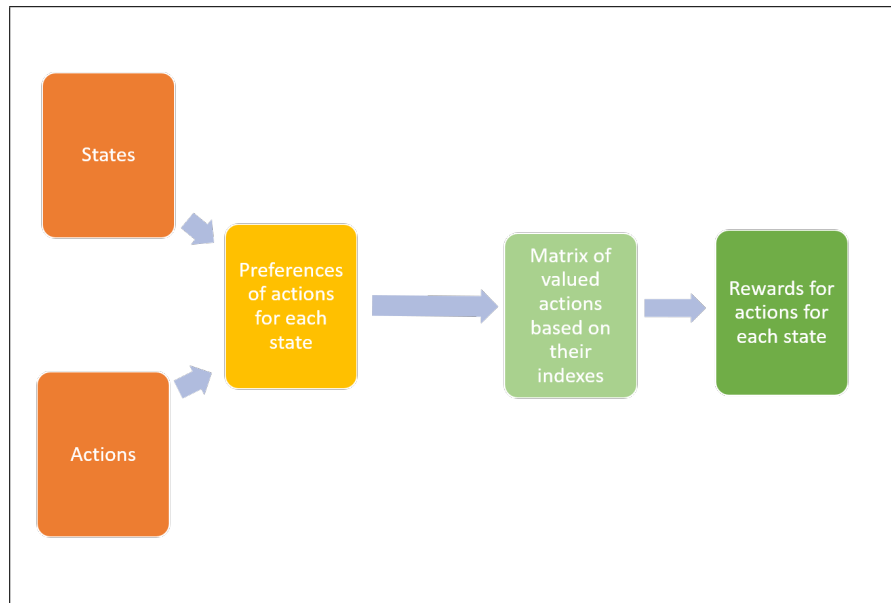


Figure 3.5: Architecture of generating reward matrix from user preferences

Preference elicitation is the next step after eliciting states and actions of the decision problem. Preferences of actions for each state are then collected from the user. States, Actions & Preferences are then stored into a JavaScript variable which is a JSON object and sent over to the Tomcat server where the above mentioned algorithm converts this JSON object into Java Collections and executes the reward function. Results are then compiled into JSON and sent back to the calling JavaScript function through an asynchronous AJAX request/response.

3.3 ELICITATION OF TRANSITIONS

As discussed earlier in the background of Markov decision processes (MDPs) on page 3 where we explain transitions of a problem domain and how they model the policy, we came up with a model to elicit transitional probabilities from the user in an intuitive way and to find symmetric actions to reduce user inputs of transitional probabilities.

Transition model describes the outcomes of each action in each state. Here the outcome is stochastic denoted by $P(s'|s, a)$, which is the probability of reaching state s' if action a is done in state s . We will assume the transitions are Markovian in the sense that the probability of reaching state s' from s depends only on s and not the history of earlier states [1].

3.3.1 CHOOSING TRANSITIONS

Transition can be described as the probability of an agent to change from one state s to another state s' . Let us say we have m number of states and n number of actions in the problem domain, then we will have m^2 transitions for every action n_1 . So we will have a total of $n \times m^2$ transition probabilities to be entered by the user. Of these $n \times m^2$ transitions, some are invalid. Those transitions which have a probability of 0 or are not possible as per the problem description are termed as **invalid transitions**. For example, in our tiger problem as described on page 41, on performing action $Listen(L)$, it has no effect on the environment which includes tiger or the person and hence the transitional probabilities of Tiger moving from left to right or vice-a-versa are invalid for action L . Hence we are only left with transitions for action $OpenLeftDoor(OL)$ and $OpenRightDoor(OR)$. Also, out of these transitions, those which don't change the state need to have a probability, since if the action does not affect the environment then the transition of being in the same state is 1. We call such transitions as “**pivots**” which are fixed and can't be removed. For example,

in the tiger problem, for action $Listen(L)$, we have two pivotal transitions: $TL \rightarrow TL$ & $TR \rightarrow TR$ since action $Listen$ has no effect or state change over the environment, we have these transitions with probability as 1.

Transitions

Choose state **transitions** in your decision-making problem.

For action:OL

TL to TL

TL to TR ❌

TR to TL ❌

TR to TR

For action:OR

TL to TL

TL to TR ❌

TR to TL ❌

TR to TR

For action:L

TL to TL

TR to TR

Let's proceed by entering **chances** for your transitions taking place in your problem

Figure 3.6: Model of Rae after removing invalid transitions for action Listen (L) of the tiger domain problem.

For action OL :

$TL \rightarrow TL : 50\%$

$TL \rightarrow TR : 50\%$

$TR \rightarrow TR : 50\%$

$TR \rightarrow TL : 50\%$

For action OR :

$TL \rightarrow TL : 50\%$

$TL \rightarrow TR : 50\%$

$TR \rightarrow TR : 50\%$

$TR \rightarrow TL : 50\%$

Thus, we observe that actions OL and OR , even if they mean different things and have different outcomes, their transitional probabilities are exactly alike and hence we can say that they have the **same effect** over the state transitions and hence we conclude that actions OL and OR are **symmetric**. By introducing *symmetry*, we can reduce the number of inputs required for transitions by almost half. So, in our Tiger Problem domain, if we have transitional probabilities for action OL and that we conclude actions OL and OR are *symmetric* then we can replicate the transitional probabilities of OL on to OR thus reducing the inputs required for action OR . We find symmetric actions by observing transitions of those actions.

Actions that have the same set of valid transitions after removal of invalid transitions, we suggest the user that those actions *could be symmetric*. We cannot always claim that actions having the same set of valid transitions are always symmetric but they can have different transitional probabilities depending on their problem domain and hence we do not enforce symmetry over the actions but rather recommend it to the user to ease their inputs.

3.3.2 ELICITATION OF TRANSITION PROBABILITIES

Once we have the valid transitions, we ask user to enter numbers ranging between 1 – 100 as chances or probability values for valid transitions. If there are any symmetric actions, as soon as the user enters their chances into the transitions, our algorithm reflects those values in corresponding transition of it's symmetric action. But, if the user believes that those actions are not symmetric and have different probability values then the user can change the value of the symmetric action to any other value and this will not reflect to the previous chance that they entered for one of the transitions of the action. For example, as we suggest that actions *OL* and *OR* are *symmetric* and as they proceed to enter chances for transitions of action *OL*, we have an immediate reflection of those values for the same transitions in the set of action *OR*. Now, if the user believes that action *OL* and *OR* are *not symmetric* which means they have different probabilities then they can change the chance value of the transition in action *OR* to any other value and the transition of *OL* will remain intact to its value.

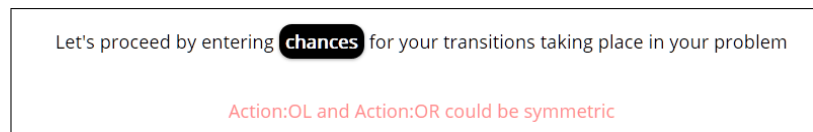


Figure 3.7: Rae suggests that action OL and OR could be symmetric.

Let's proceed by entering **chances** for your transitions taking place in your problem

For Action:OL

TL to TL %

TL to TR %

TR to TL %

TR to TR %

For Action:OR

TL to TL %

TL to TR %

TR to TL %

TR to TR %

For Action:L

TL to TL %

TR to TR %

Action:OL and Action:OR could be symmetric

Figure 3.8: User enters the chances of transitions they chose and symmetry function helps to fill in values for symmetric actions. In this case, when user enters values for action OL, Rae fills values for action OR simultaneously.

This is a transition function. Again, ignore if you are unable to understand, this is how I see your perspective of transitions.

OL	TL	TR
TL	0.5	0.5
TR	0.5	0.5
OR	TL	TR
TL	0.5	0.5
TR	0.5	0.5
L	TL	TR
TL	1	0
TR	0	1

Figure 3.9: Rae generates transition matrix in POMDP format and associates chance probabilities entered by user into transition values.

Algorithm 2 Algorithm for generating transitions

```

1: procedure SHOW_TRANSITIONS(states,actions)
2:   for action in actions do
3:     print("For action": action)
4:     for  $i = 0$  and  $i < states.length$  and  $i ++$  do
5:       for  $j = 0$  and  $j < states.length$  and  $j ++$  do
6:         if  $i == j$  then
7:           Show transition as pivotal
8:         else:
9:           Show transition as non-pivotal ▷ This will display a cross mark in front
           of the transition to remove if invalid
10:    return: ( $state[i] \rightarrow state[j]$ )

```

Algorithm 2 for generating transitions proceeds on the call to function on line 1 of the algorithm 2. This call is made as soon as the user is done entering preferences of their actions for the states. Line 1 in the algorithm takes *states*, *actions* as inputs. From line 2, the program outputs a transition for every action from one state to another. If the transition is from one state to the same state then it is a pivot and the transition is displayed without pivotal-icon so that the user won't be able to remove the pivotal-transition.

Algorithm 3 Algorithm for removing invalid transitions

```

1: procedure REMOVE_TRANSITIONS(transitions)
2:   delete transition ▷ Remove non-pivotal transitions

```

Algorithm 3 is for the function on line 1 which is called when the user removes the transition which according to him is an invalid-transition.

Algorithm 4 Algorithm for suggesting symmetric actions

```

1: procedure SUGGEST_SYMMETRY(valid transitions)
2:   Initialize : action1, action2, temp1, temp2
3:   for  $k = 0$  and  $k < actions.length$  and  $k ++$  do
4:      $action1 \leftarrow actions[k]$ 
5:      $temp1 \leftarrow transitions[action1]$ 
6:     for  $i = k + 1$  and  $i < actions.length$  and  $i ++$  do
7:        $action2 \leftarrow actions[i]$ 
8:        $temp2 \leftarrow transitions[action2]$ 
9:       if  $temp1 == temp2$  then ▷ Compare set of transitions for actions
10:         $action1$  &  $action2$  are symmetric
11:   return:  $action1$  and  $action2$  could be symmetric

```

Algorithm 4 is for the function on line 1 which is called when user submits his set of valid-transitions that are left after removing invalid-transitions. Once, we have the set of valid transitions, we try to find symmetric actions as explained in section 3.3. Lines 2 proceeds to create temporary variables to store transitions for every action to compare with transitions of every other action. If we have same set of valid-transitions for two actions then call those two actions as symmetric. Algorithm proceeds with set of same transitions within all the pairs of actions. We put symmetric actions in the JSON variable where we store states, actions, transitions, and observations.

Algorithm 5 Algorithm for filling transitions of symmetric actions

```

1: procedure FILL_SYMMETRY(actions,symmetry,transitions)
2:   if  $action1$  and  $action2$  are symmetric then
3:     Copy transitional values from action1 to action2
4:     Set action2 visited ▷ This is to ensure that if action1 and action2 are not
       symmetric then changing values of action2 won't affect action1

```

Algorithm 5 is called when the user start entering values in the input-boxes for entering chances of the transitions. Whenever the user enters any value in the input box, function on line 1 checks if the action associated with that transition has any symmetric action. If it does, then it copies the value that user enters in the action to its symmetric sibling action.

Once they are done, symmetric sibling changes its flag so that when user changes value of the symmetric sibling action (assuming the user believes those two actions are not symmetric) then the values aren't copied to the original action making sure that *symmetry* is a **suggestion** and not enforcing on the user's model of problem domain.

3.3.3 ARCHITECTURE OF ELICITING TRANSITIONS

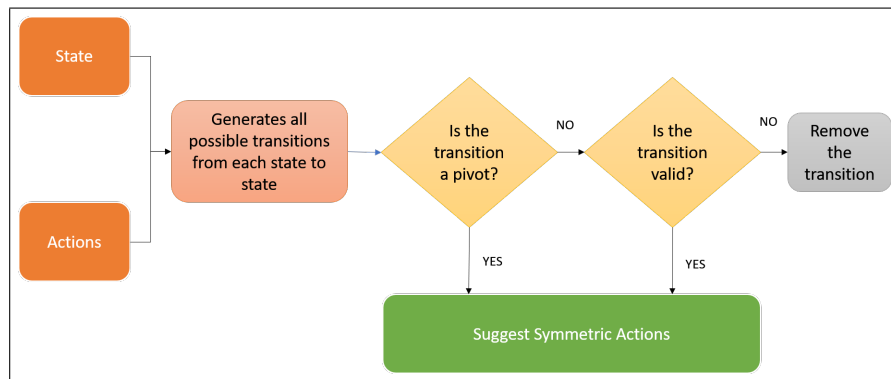


Figure 3.10: Architecture of eliciting transition function of the domain problem

For eliciting transitions from the user for their problem domain, we need *states & actions* from the user which are stored in the JavaScript array. We generate all the possible transitions from each state to another for every action along with clickable buttons to let the user remove invalid transitions. If the transition is a pivot then it is added to a stack of valid transitions. If the transition is invalid, user removes it. Finally, all the transitions that are left after removing invalid transitions are added to the stack of valid transitions. Once, we have all the valid transitions, we calculate symmetric actions as explained in the elicitation methodology and present symmetric actions to the user and also to the calling function which will copy probability values of transitions to their symmetric siblings.

When all the probabilities have been entered for valid transitions, a validation function checks to make sure the total of transition probabilities is correct and transition matrix are generated and presented by Rae.

3.4 ELICITATION OF OBSERVATIONS

In partially observable markov decision processes (POMDPs), the state's are not completely visible and hence the only way to know about current state or any other state is to have information through agent's sensors that we call as **observation**. For example, in the tiger problem which is described on page 41, we can hear the tiger's *growl from the left* or *growl from the right* after performing action *Listen(L)*. Thus, we have two observations - *Growl from the left (GL)* and *Growl from the right (GR)* which informs the agent about the tiger's current state on choosing to perform action *Listen(L)*.

What **observations** do you get to help you in determining the state of your decision problem?
 Enter these observations as comma-separated labels in the following input box.
Observations:

Choose one or more actions through which you get your observations.
 Please select action-choice/choices

You selected 'L' as action-choice(s) for observation variables

On performing L, from which state(s) could you possibly generate 'GL'? Enter those state labels below.

On performing L, from which state(s) could you possibly generate 'GR'? Enter those state labels below.

Figure 3.11: User enters observation labels and chooses an action that gives observations.

We can see the observation matrix for tiger problem on page 11. It is represented as $O(s, a, z)$ where z is the observation $z \in z_{GL}, z_{GR}$, a is the action that gives us observations and s is the state that is informed after observing z . Thus, observation matrix is a probability distribution over observation variables for that state. If the agent observes GL then the tiger is probably behind the left door. We say *probably* because we still don't know the exact state of the tiger and observations are noisy in nature and hence cannot have a value of probability 1. Hence, if we check the observation matrix of the tiger problem, we see that the probability of TL , given GL on performing action L is 0.85 which means there is 85% chance that the tiger is behind the left door. Which also means, there is 15% chance of tiger being behind the right door. We can have same inference for Growl from the right GR where the tiger is behind the right door with a chance of 85% while it could be behind the left door with a chance of 15%.

We wanted to have as many less inputs from the user as possible and hence we came up with a technique of generating probability distribution over the states based on the observations without asking for any probability values but to rather use preference elicitation model for the observations. Although, we state to use preference elicitation model for observations, the model we use here is completely different than our Reward function or AHP [7] process. User is asked to enter states in an order of higher observability from left to right for a particular observation on performing a specific action. Thus, in case of Tiger problem, user is asked to enter states of Tiger problem that he believes would be observed on listening to growl from the left (GL) on performing action Listen (L). User would enter:

On performing L, from which state(s) could you possibly generate 'GL'? Enter those state labels below.

TL

On performing L, from which state(s) could you possibly generate 'GR'? Enter those state labels below.

TR

Figure 3.12: State labels entered through which the user observes the specified observation variables

OL	TL	TR
GL	0.50	0.50
GR	0.50	0.50
OR	TL	TR
GL	0.50	0.50
GR	0.50	0.50
L	TL	TR
GL	0.90	0.10
GR	0.10	0.90

Figure 3.13: Observation matrix for only single states through which the user observes the observation-variables

Users may also enter:

On performing L, from which state(s) could you possibly generate 'GL'? Enter those state labels below.

TL,TR

On performing L, from which state(s) could you possibly generate 'GR'? Enter those state labels below.

TR,TL

Done

Figure 3.14: Multiple state labels entered through which the user observes the specified observation variables

Which means after observing GL, the user may believe the tiger is behind the left door but given that their observation was noisy, the tiger might be behind the right door. This ranks TL a little higher than TR but the probability value that TL receives would be lower than the value it would have received when it was the only state user believed to have observed.

OL	TL	TR
GL	0.50	0.50
GR	0.50	0.50
OR	TL	TR
GL	0.50	0.50
GR	0.50	0.50
L	TL	TR
GL	0.60	0.40
GR	0.40	0.60

Figure 3.15: Observation matrix for multiple states through which the user observes the observation-variables

Algorithm 6 Algorithm for generating Observation matrix

```

1: procedure GENERATEOBSERVATIONMATRIX(states, actions, observations, observa-
   tion_actions,elements) ▷ elements refer to entered states for
   observations
2:
3:   for action in actions do
4:     observation_weights ← ▷ Initialize
5:
6:     for state in states do ▷ Initialize
7:       observation_weights[action][state] ←
8:
9:       for observation in observations do
10:        if observation_actions.indexOf(action) >= 0 then
11:          if elements.indexOf(state) >= 0 then
12:            index ← elements.indexOf(state)
13:            val ← index × ((2 × (states.length) - (elements.length)) ×
   ( $\frac{1}{\text{observations.length}}$ ))
14:            observation_weights[action][state] ← val
15:            totalOfObservations ← (totalOfObservations + val)
16:          else
17:            val ←  $\frac{1}{\text{states.length}}$ 
18:          else
19:            val ←  $\frac{1}{\text{states.length}}$ 
20:            totalOfObservations ← (totalOfObservations + val)
21:          if totalofweights ≠ 1 then
22:            reduceWeight ←  $\frac{1}{\text{totalOfObservations}}$ 
23:            for observation in observations do
24:              tempVal ← observation_weights[action][state][observation]
25:              tempval ← (tempval × reduceWeight)
26:              observation_weights[action][state][observation] ← tempVal

```

Algorithm 6 proceeds when the user has entered all the states for observation variables for the action that gives the user their observations, function on line 1 is called. Function receives state, action and observation labels, observation-action label that user chooses which provides the observations, elements- which refer to the user's preference of states through which they observes a particular observation label for that action which provides them the observations.

Line 13 calculates a normalized value for action, state and observation. If the state is present in the set of observation preferences then it calculates the value based on its index value as described in the section 57.

3.4.1 ARCHITECTURE OF ELICITING OBSERVATIONS

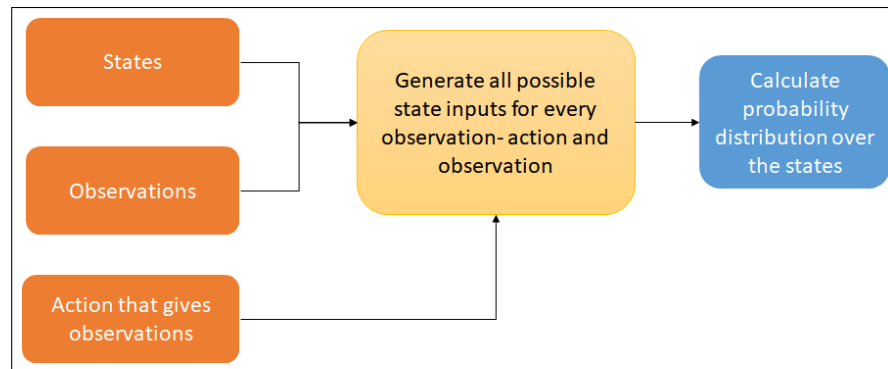


Figure 3.16: Architecture of eliciting observations of the domain problem.

For elicitation of observations from the user for their domain problem to observation function of the POMDP, we require *states*, *observation labels* & *action that gives observations* which are stored in the JavaScript variable. JavaScript function generates all possible state inputs for every observation-action and observation. Following the algorithm for generating observation probabilities, the observation matrix is presented by Rae.

3.5 INTERACTIVE POLICY

Once the user completes entering states, actions, preferences, transitions and observations, a POMDP file is generated which is then sent to Sarsop - A POMDP solver which evaluates the policy and generates policy file in `.dot` file. We designed an interpreter which converts this `.dot` file into JavaScript objects and sends to D3.js API which then generates an interactive `svg` tree which can be explored.

Since the policy is a cyclic graph which means that once the goal state has been reached (in Tiger problem, if the policy reaches to OL or OR), it resets the belief to the same state it began with. Thus, this is a cyclic graph and since we need to represent our policy in a tree structure, we will have an infinite tree. To represent such an infinite tree which would go in an infinite loop and never generate a tree, we designed it in such a way that when any of the actions are clicked by the user, only two levels further down that action are explored and added into the D3.js tree. Thus depending on the user interaction, new nodes and edges are generated through our interpreted node-edge structure from the `.dot` file. Hence, we never explore the whole tree which would be a deadlock and hence we save memory and processing by implementing this feature.

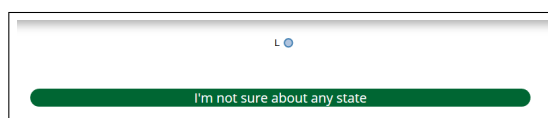


Figure 3.17: Rational policy suggested by Rae based on user inputs. This is an interactive policy it has suggested the first action Listen (L).

Observe that the action hasn't been clicked yet, and hence the probability of Tiger behind one of the doors is not known (50%).

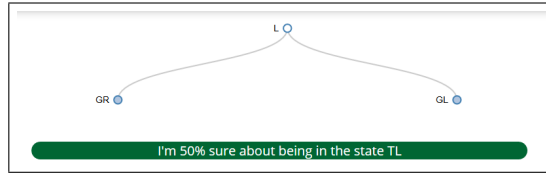


Figure 3.18: After clicking action L, policy asks to choose what we observe.

Now, after choosing the action, state probability is reflected to 50% and the policy asks user to choose one of the observations that he observes.

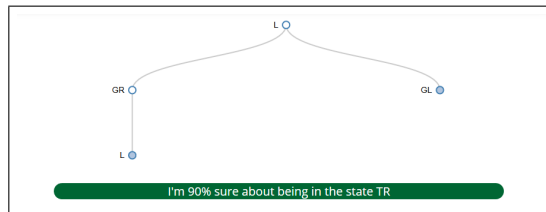


Figure 3.19: Choosing observation 'Growl from the right' (GR), policy suggests next action which is L.

After choosing action L as suggested by the policy, state reflects the possible change of the previous observation to update Tiger's location to the right with a probability of 90%.

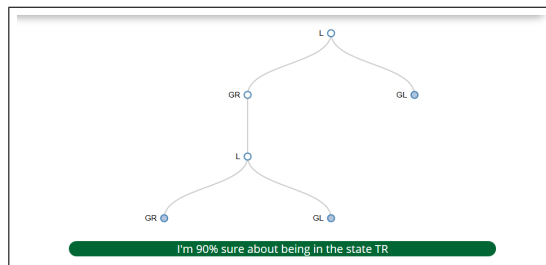


Figure 3.20: Choosing action L after the policy recommendations.

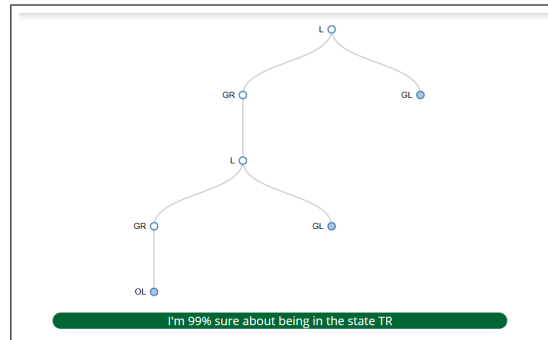


Figure 3.21: After the user chooses action L in the previous policy, he is asked to choose one of the observations.

If the user chooses GR, then we see that the state probability of Tiger being on the right becomes 99% and the policy recommends to open the left door (OL).

3.5.1 ARCHITECTURE OF INTERACTIVE POLICY

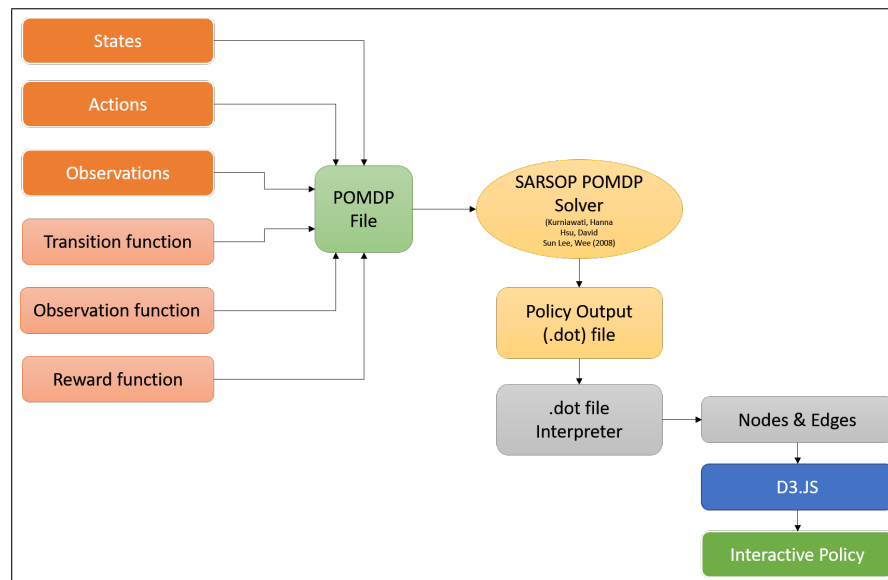


Figure 3.22: Architecture of interactive policy

To present an interactive policy to the user, we need all the variables and probabilities elicited from the user. We get *states*, *actions*, *observations*, *transition function*, *observation function* & *reward function* generated by Rae from JavaScript variable which is a single

variable that stores all in the information in JSON array. We send this as a JSON object to the server which returns a **POMDP file** which is in specific format as specified by Anthony Cassandra (1998) [4].

This file is then read by server which executes SARSOP [5] C++ code to generate policy file which is a **(.dot)** file. This file is then read by JavaScript function to generate *nodes&edges* which are stored in JavaScript variables. **D3.JS**, a JavaScript graph library constructs an interactive graphs by accessing the nodes and edges at a level of 3. Since, the policy could be a circular graph, representing such a graph as a tree would only be a deadlock on any computer system as the tree would be spanned infinitely since it is a circular graph. Hence, we only explore nodes at three levels, so whenever the user interacts with the policy and explores a specific path by clicking on the nodes, the next level nodes of only that path are added into the **D3.JS** JavaScript function. Thus, this method ensures that any policy can be represented as an interactive policy.

CHAPTER 4

EXPERIMENTAL ANALYSIS

In this chapter, we discuss the hypotheses regarding our agent on human usability study and results of the survey evaluation for our model.

4.1 HYPOTHESES

In this section, we present our hypotheses and provide correlations to analyze our data and to see how the data answers to our hypotheses.

Hypothesis 1: *Rae is an effective interactive decision aid that can satisfactorily elicit a decision problem as a POMDP.*

We asked questions regarding the policy and our agent to understand whether if they are satisfied with working of Rae in Survey questionnaire 2 as represented in Appendix B, recommended policy for their decision problem and also their suggestions to improve the agent.

Effectiveness of an interactive decision aid can be found based on the score that we receive from participants based on evaluation of effectiveness of Rae. We have the following statistics:

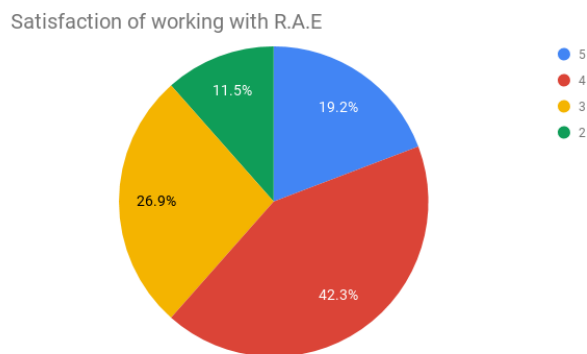


Figure 4.1: This graph is based on scores received from participants based on satisfaction of working and interacting with Rae in Survey: 2 as described in Appendix B

In Figure 4.1, we have a scale of 1 to 5 where *1: Least satisfied* and *5: Very satisfied*, we observe from the above graph that 19% of the participants were very satisfied with working and interacting with Rae, 42% of the participants were satisfied with the working of Rae, 27% of the participants gave a score of 3 while 11.5% participants were not satisfied enough who gave a score of 2. We don't see any participant who was least satisfied with working of Rae.

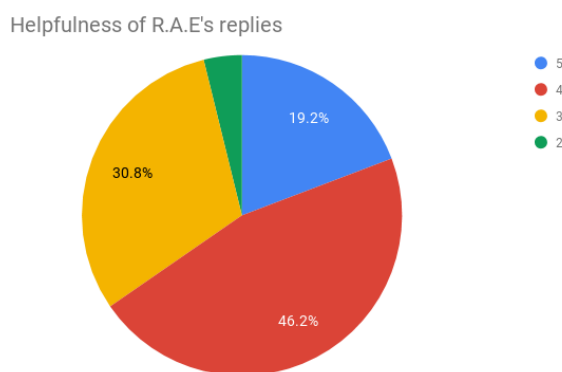


Figure 4.2: This graph is based on scores received from participants based on interaction with Rae and their scores about effectiveness based on a question in Survey: 2 as described in Appendix B

In Figure 4.2 we have a scale of 1 to 5 where *1: Least satisfied* and *5: Very satisfied*, we have 19.2% of the participants who gave a score of 5 to the interaction of Rae, 46.2% participants gave a score of 4. These scores tells us how effective were the Rae's replies

and based on the above analysis we can conclude that interaction with Rae, which includes entering states, actions, transitions, preferences, observations, interacting with policy, and asking questions to Rae through voice/text based input and receiving answers from Rae. We can conclude that our model of Rae is effective in terms of working and interaction.

To understand whether a decision aid can satisfactorily elicit a decision problem as a POMDP, we will analyze the data to first understand how closely the participants were able to relate the decision problem they were asked to solve using Rae and we will analyze the data to see how much score the suggested policy received from the participants.

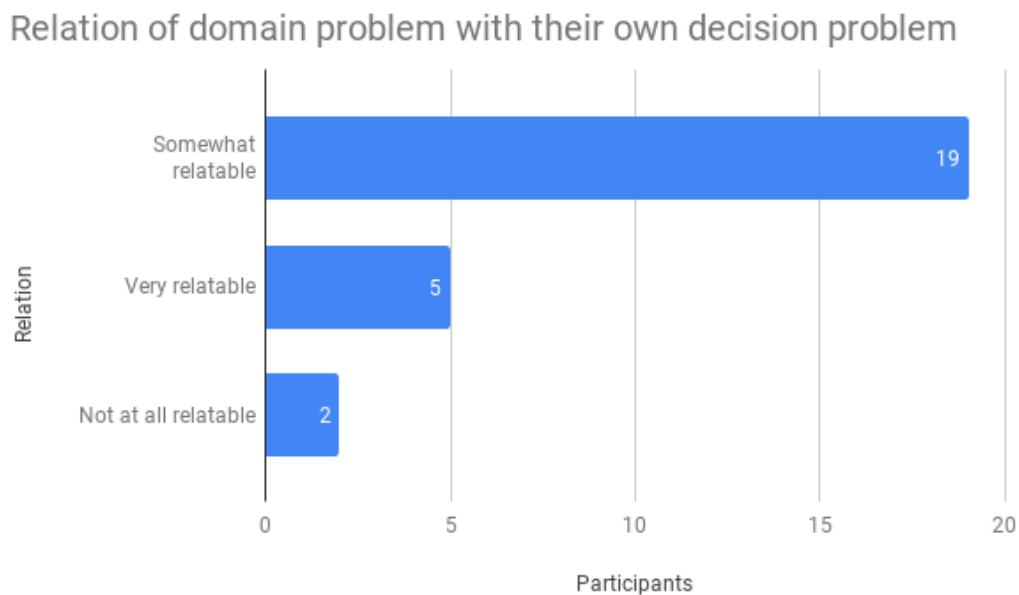


Figure 4.3: This analysis is based on relation of the problem domain with their own based on a question in Survey: 2 as described in Appendix B

In Figure 4.3, we asked a question to understand how much the participants could relate the 'Change of Major' domain problem with their own and they were given three option choices: *Somewhat reliable*, *Very reliable*, *Not at all reliable*. We observe that 5 of the total participants found this domain very reliable which means they were thinking

about changing their major. 19 of the participants found it somewhat relatable. Therefore, 92% of the participants were able to relate to this domain problem that they solved using Rae.

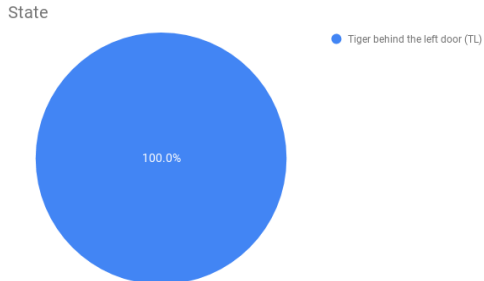


Figure 4.4: Survey-1 (State): Analysis of data we collected from Survey questionnaire -1 which asks participants to select state from the available choices.

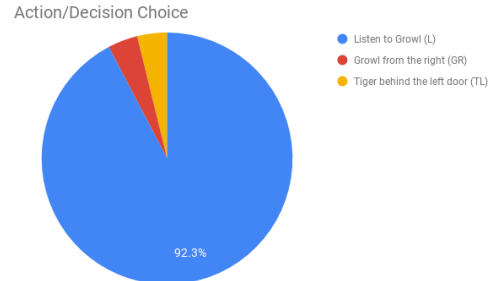


Figure 4.5: Survey-1 (Action): Analysis of data we collected from Survey questionnaire -1 which asks participants to select action from the available choices.

Figure 4.4 and Figure 4.5 are the analysis of data we collected from Survey questionnaire -1 as given in the Appendix A which asks participants to select state and action from the available choices. We observe that all the participants correctly chose state while 92% of the participants chose the correct action from the available option choices. Based on these results, we can say that the participants had a clear understanding of state and action definitions before exploring the policy.

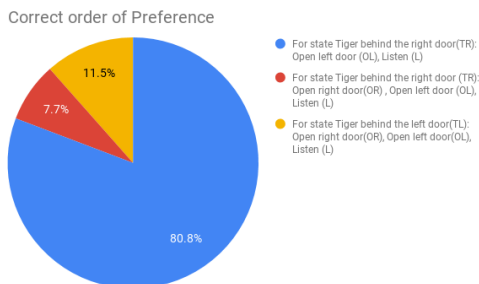


Figure 4.6: Survey-1 (Preference) is based on the understanding of preferences that are asked to generate reward matrix.

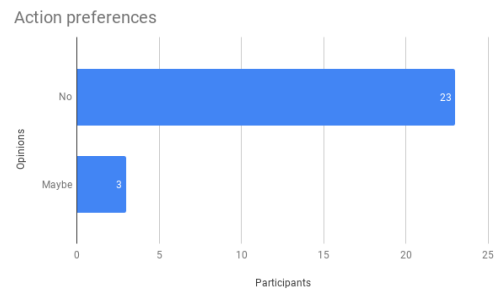


Figure 4.7: Survey-2 (Preference) is based on the question asked in survey questionnaire-2 which asks whether the participants believe their preferences of actions should have been different.

We observe that 80% of the participants were able to understand preferences of actions for each state before exploring the policy. After exploring the policy, participants could change their preferences or transition probabilities or observations which would reflect a new policy. In Survey questionnaire-2 which is shown in the Appendix B, we ask them if they believe their order of preferences should have been different and we observe that 23 out of 26 participants answered no which is 88% of the participants. Hence, we can conclude that the participants understood the idea of preference of actions for each state.

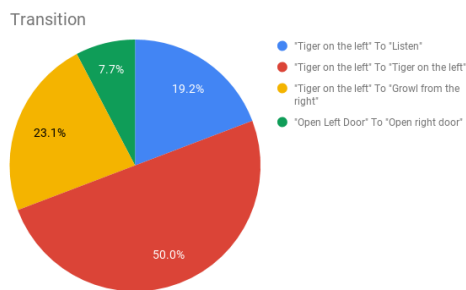


Figure 4.8: Survey-1 (Transitions) is based on the understanding transitions.

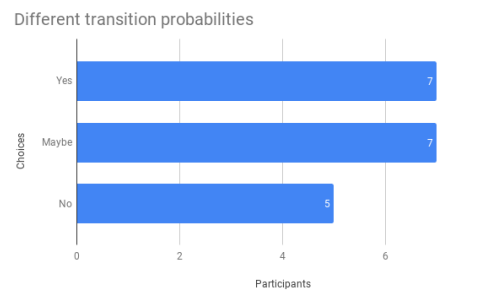


Figure 4.9: Survey-2 (Transitions) is based on the question asked in survey questionnaire-2 which asks whether the participants believe their transitions should have been different.

We observe that 50% of the participants chose the correct option for the question asked in Survey questionnaire 1 in Appendix A about choosing a correct transition option choice. In Figure 4.8, we see that 14 participants believe their transition probabilities should have been different based on the policy they interacted before answering to this question in Survey questionnaire 2 in Appendix B. We can conclude that the idea of transitions needs to be explained in a better way. And we also believe that coming up with probability values in the range of 1-100 is dependent on the user's perspective and judgment.

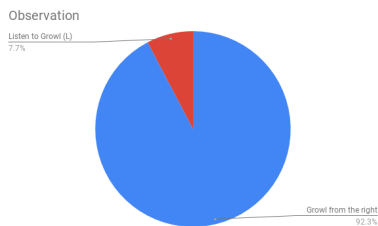


Figure 4.10: Survey-1 (Observation) is based analysis of the question asked about choosing observations before exploring the policy.

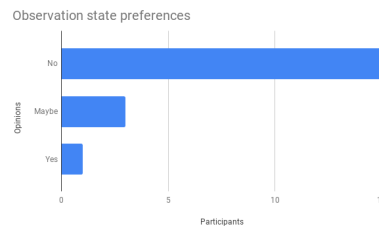


Figure 4.11: Survey-2 (Observation) is based analysis of the question asked about choosing observations after exploring the policy.

We observe in Figure 4.10 that 92.3% of the participants were able to choose the correct observation from different choices. In Figure 4.11, we see that most of the participants believe their order of states for observed labels for action that gives observations should not have been different. We can conclude that participants understood the meaning of observations and also were satisfied with the preferences of states for observations.

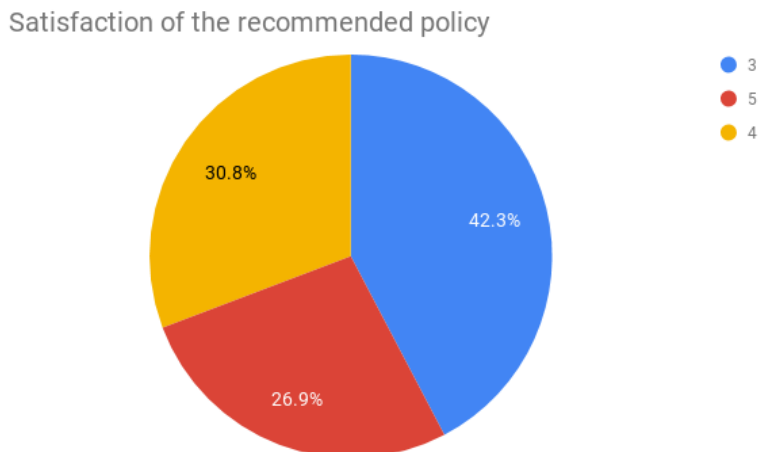


Figure 4.12: This analysis is based on analysis of policy which tells us about participants' satisfaction with the recommended policy by Rae in Survey: 2 as described in Appendix B

In Figure 4.12, we asked a question regarding policy satisfaction and participants chose one of the scores of the scale 1-5 where 1: *Least satisfied* and 5: *Very satisfied*. We observe that 27% of the participants were very satisfied, 31% of the participants gave a score of 4

which means they were satisfied as well. 42% of the participants gave a score of 3 which means they were somewhat satisfied with the recommended policy. Rae didn't receive a score of 2 or 1 which indicates that the policy was at least satisfactorily acceptable by the participants.

Based on the above analysis of our collected data, we can conclude that our hypothesis that *Rae is an effective interactive decision aid that can satisfactorily elicit a decision problem as a POMDP* holds true.

Hypothesis 2: *Rational policies obtained by solving acceptable POMDPs are utilitarian prescriptions.*

To evaluate the rational policies calculated by Rae on solving POMDPs generated by eliciting user decision problem, we asked questions regarding their satisfaction with the suggested policy.

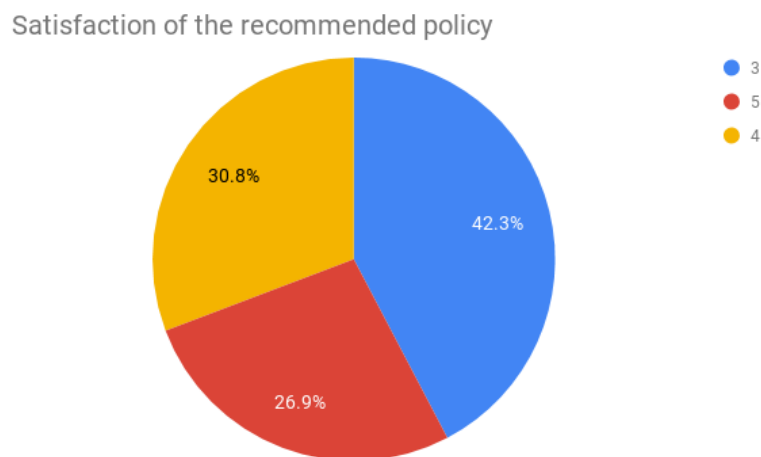


Figure 4.13: This analysis is based on analysis of policy which tells us about participants' satisfaction with the recommended policy by Rae in Survey: 2 as described in Appendix B

In Figure 4.13, we observe that all the participants were satisfied with lowest score of 3 and highest score of 5 on a scale of 1 to 5 where *1: Least satisfied* and *5: Very satisfied*.

Thus, the policies suggested by Rae were satisfactorily acceptable according to the participants. We can thus conclude from this analysis that Rational policies obtained by solving acceptable POMDPs are utilitarian prescriptions for the participants' decision problem.

Hypothesis 3: *As a demonstration of the external validity of the individual differences in probability assessment, modal transitional probabilities will correlate with probabilities elicited in a judgment problem with unrelated content.*

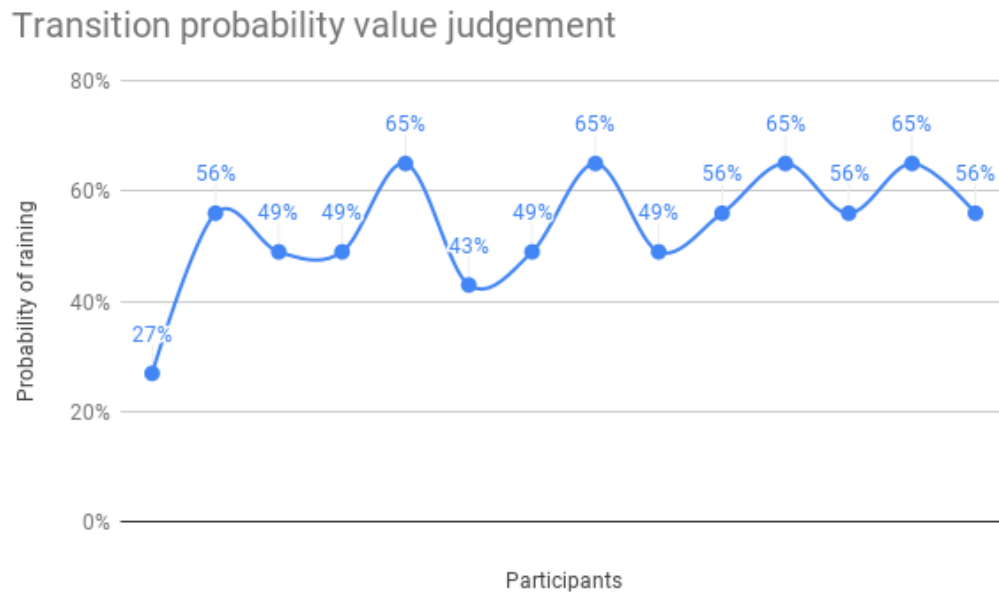


Figure 4.14: This analysis is based on the question asked in Survey-2 as described in Appendix B regarding probability judgment of the participant.

In Figure 4.14, we observe a line graph of Participants against probability of raining. Refer to the question no:3 in Survey-2 in Appendix B. The participants' answers were not related to the domain problem that they solved using Rae. We observe that participant's chose probability of raining based on their experience. We can call this as *Response bias*.

According to Furnham [14], *Response bias* which is a generic term for a whole range of responses to interviews, surveys or questionnaires which bias the response (from the correct, honest, accurate response). They include the social desirable or faking-good response as well as its opposite faking-bad (or mad), acquiescence or yea-saying (the tendency to agree irrespective of the question) or its opposite or nay saying, extremity response set (always choosing extreme opposites) or its opposite, mid-point response set etc. *Social desirability* as defined by Nederhof (1985) in the paper published by Furnham [14] is a tendency to deny socially undesirable traits and to claim socially desirable ones, and the tendency to say things which place the speaker in a favorable light.

The recommended policy suggested by Rae depends on the participant's understanding of the domain problem. Thus, the policy depends on the participant's choice of preference of actions, transition probabilities, and observations and these values are unique to every participant. Our goal is to find if *Response bias* or *Social desirability* has any effect on the recommended policy. Since, probability entered for transitions range in between 1 to 100 and to enter such numerical values, one has to identify the domain completely and if they don't then the values that they enter may involve some of the response bias or social desirability. Our objective is not to find if the participants had response bias but we tend to find if they have any effect on the transition values. We calculated mean, median and modal values of the transitions for each participant who attempted the question of rain as described in the Survey 2 in Appendix B. We calculated correlation between the answers to the question asked in Survey 2, which is the independent data of participants as it has no relation with the problem domain they solved using Rae, which was correlated with mean, median and modal values of the transition probabilities that participants entered in the chance input-boxes of Rae.

Table 4.1: Table representing Pearson's Bi-variate one-tailed correlations calculated in IBM SPSS software between the survey question and mean, median & modal values of the transition probabilities

	Transition probabilities		
	Mean	Median	Mode
Judgment question	0.155	-0.110	-0.507
Significance	0.299	0.354	0.032

We observe from the correlation table 4.1 that we have a positive correlation between the judgment probability answers and the mean of transition probabilities of participants although not much significant which means it could be because of the chance. But, we observe that there is a negative correlation between the judgment probability answers and modal value of the transition probabilities of the participants which is significant at 0.05 level (1-tailed). Which means that there is an inverse correlation between the modal transitional values and the probability judgment values. Although, the negative correlation is not close to -1 and greater than less than -0.5 which means, we can say that there is a linear correlation in which modal transitional value decreases on increase of judgment values. Significance is 0.032 means this is not because of the chance and there is a strong correlation between the modal values of transitions and the answers to the judgment question. We may conclude from this that *response bias* may be affecting transitional probability values based on their judgment. For example, a participant may feel confident to enter higher transitional probability values for their transitions but may enter lower values for the judgment question for social desirability.

Hypothesis 4: *Participants who relate strongly to the problem, although they stand to benefit more from policy-based decision aids, may also have more contradictory thoughts about the topic. It is therefore hypothesized that, although there is no direct measure of contradictory thoughts, relatability of the problem will correlate with policy satisfaction.*

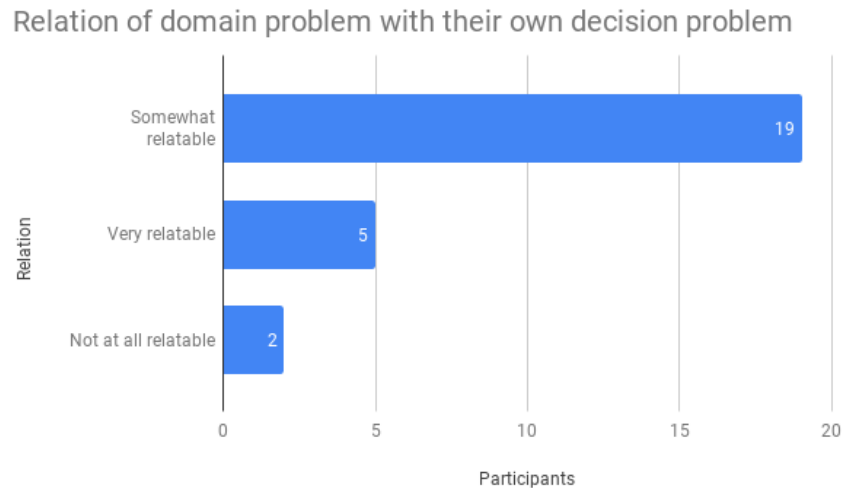


Figure 4.15: Survey: 1, Q: 7, Relation of 'Change of Major problem' to their personal decision problem.

We observe from Figure 4.15 that most of the participants found the domain problem to be somewhat relatable while 5 of the total participants found the domain problem very relatable.

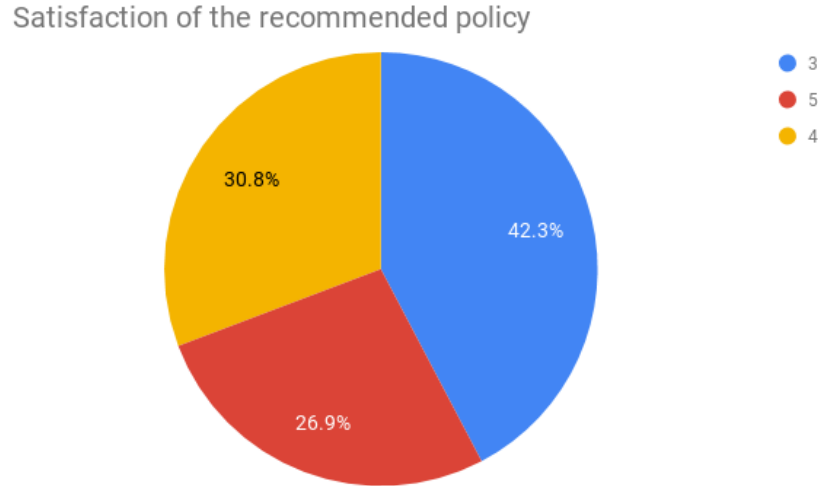


Figure 4.16: Survey-2, Q:1, Satisfaction of the recommended policy.

In Figure: 4.16, we observe that most of the participants were satisfied with the recommended policy by Rae. Scale used for this parameter was from 1 to 5 where 1: *Least Satisfied* and 5: *Very Satisfied*. We didn't receive a score of less than 3 for policy satisfaction which explains that participants accepted the policy satisfactorily as a recommendation by Rae.

We try to find correlation between the *Relation of domain problem with their personal decision problem* and *Satisfaction of the recommended policy by Rae*, to find if we can prove the Hypothesis.

Table 4.2: Table representing Pearson's Bi-variate one-tailed correlations calculated in IBM SPSS software between the relation of problem domain and policy satisfaction.

	Relation to problem domain
(Pearson's correlation constant)	-0.329
Significance (One-tailed), $p < 0.05$	0.05

Thus, the correlation result suggests that there is a correlation between the relation of the domain problem to their personal decision problem and policy satisfaction. A negative correlation suggests that participants for whom the domain problem was very relatable found the policy less satisfactory than the participants to whom the domain problem was somewhat or not at all relatable. We can infer from the results of this correlation that the participants who were able to relate the domain problem to their personal decision problem had a better understanding of the resulting policy and the expectation and were less satisfied with the recommended policy, given their expectations and participants with less relatability of the problem domain were not clear about the expectation of the policy and found the recommended policy more satisfactory. Hence, we can conclude that our results support the said hypothesis.

Hypothesis 5: *Modal transition probabilities and policy satisfaction are correlated and it reflects the response bias of participants associated with a correlation between individual differences in probability assessment and modal transition probabilities in policy satisfaction.*

This hypothesis tries to explore the possibility if the *Response Bias* [14] as discussed on the page: 75 has any effect on the recommended policy. We try to find the correlation between *Modal Transition Probabilities* and *Policy Satisfaction of the participants*. **Modal Transition Probabilities** are calculated by taking all the transition probabilities of each participant and finding the modal value. Modal value suggests whether the participant had any response bias by finding if they had extreme probability values in their transitions. By extreme we mean either close to 100 or close to 0. This suggests whether the participant had any effect on the transitions and if this reflects in the recommended policy.

We couldn't find any correlation between the *Modal Transition Probabilities* and *Policy Satisfaction*. We don't exactly know the reason but we believe that this could be because of the less amount of participants. Thus, our results conclude that they don't support the said hypothesis and we couldn't find correlation to support it.

4.2 HUMAN USABILITY STUDY

Human usability study was setup in two separate rooms at Department of Psychology. One room was designed to have participant use one of the computers where this model was active and have no disturbance during the study. We had setup our servers on a different pc in another room and their interaction with the model and activities on the the pc were monitored to ensure there was no error or problems for which the participant may be confused.

Study proceeds by giving participants a consent form which explains them about the study and their rights and consent for the study. A briefing form is then presented which explains the tiger problem. It also explains what are different states, actions and observations from the tiger problem definition and we explain transitions and their probability values. Once the student has read through the briefing form, a few questions regarding their understanding of 'preferences' and 'transitions' are asked and clarified.

Participant is then presented with our model on the computer which presents a "Change of major" decision problem. Only the description is presented without any explicit definitions of states, actions, transitions or observations related to that problem. It is expected from the participant to relate this problem with tiger problem domain and to come up with their own states, actions, observations, transitions, etc. from the problem domain definition thus making the decision problem unique to every participant since they are free to choose states as per their actual current majors and interests.

Participant proceeds to enter their own states, actions, preferences, transitions, and observations. We have a voice/text based assistant which is ergonomically placed at the bottom-right corner of the screen which can also be accessed through labeled words throughout the model. This assistant helps participant in clearing definitions or understanding of parameters such as states, actions, chances, symmetry, transitions, etc. by interacting with the

text/voice assistant. Once, the participant is done entering states for their observations, they are presented with the first Survey form. This survey form asks questions to know whether or not if they understood the definitions of states, actions, observations, transitions and if they could relate to this problem domain.

Once the participants complete first survey form given in the Appendix A, they are presented with an interactive policy starting from the first suggested action as a clickable node. Participants explore the policy by making choices between their observations as a make-believe scenario. If the participant isn't satisfied with the policy or wishes to change a few parameters, they are allowed to do so and they may change their preferences, transitional probabilities, observations, etc and explore the updated policies. Once the participant is satisfied with the policy or wants to proceed, they are presented with the second survey questionnaire presented in Appendix B.

The second questionnaire asks questions to participant regarding the policy, their satisfaction of the policy, and overall interaction with the agent and model evaluation. This questionnaire gives us a thorough evaluation of the model and we will explain our results and analysis based on the data that we collected from survey questionnaire-1 and survey questionnaire-2.

4.3 DATA ANALYSIS

We present our data analysis of 27 participants who were studying 'Introduction to Psychology' course at Department of Psychology.

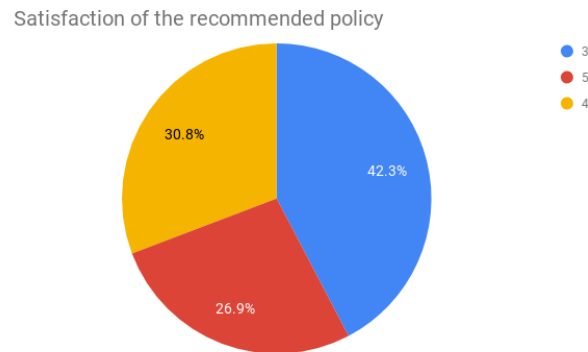


Figure 4.17: Survey-2 (Policy Satisfaction) is based analysis of policy which tells us about participants' satisfaction with the recommended policy.

In the graph in Figure 4.17, we have a range of choices from 1-5 where 1 : *Very unsatisfied* and 5 : *Very satisfied*, we observe that 27% of the participants were very satisfied with the policy who gave a score of 5 to the question about policy satisfaction. Participants who gave a score of 5 to this question were not evaluated for questions regarding states, actions, transitions or observations. We asked them questions in survey questionnaire 2 in Appendix Bonly regarding the evaluation of decision-making agent Rae.

Participants who answered less than 5 to this question were considered as participants who were not satisfied with the recommended policy and were asked questions regarding their dissatisfaction to understand what went wrong or what should have been better.

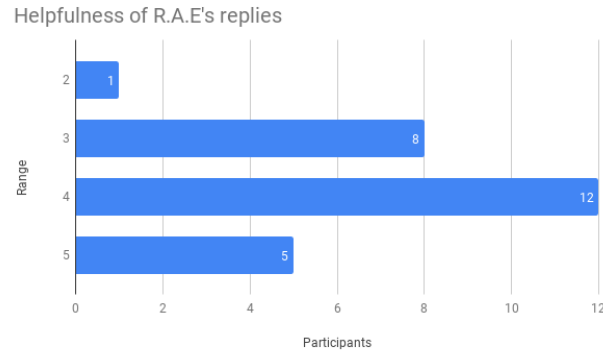


Figure 4.18: Survey-2 is based on analysis of how helpful were the Rae's replies and aid during the process until policy exploration.

In Figure 4.18, we observe that the Rae's replies to the questions asked by participants were satisfactory. Rae replied to their questions through voice/text output and asked if the participants wanted examples based on the provided definition or answer to their question. Examples were based on Tiger domain problem as explained on page 41, and clarified their understanding of the concept their question was based on.

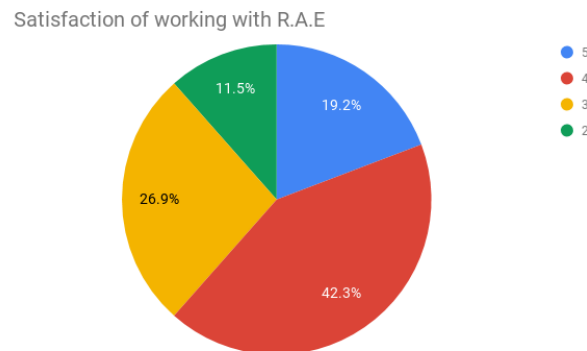


Figure 4.19: This analysis is based on analysis of satisfaction of working with Rae.

In Figure 4.19, we have a range of 1-5 where 1: *Least satisfied* and 5: *Very satisfied*. We observe that 42.3% of the participants gave a score of 4 and 19% of the participants gave a score of 5. Thus we can conclude that the participants were satisfied with the working of Rae.

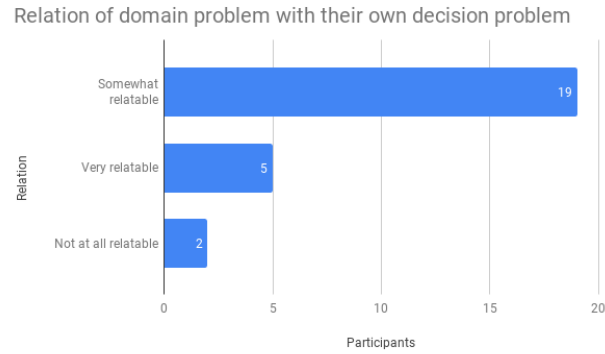


Figure 4.20: This analysis is based on relation of the problem domain with their own.

In Figure 4.20, we asked participants on how much were they able to relate their decision problem of Change of Major with the problem domain that we defined which they had to use Rae for solving the problem and evaluating the recommended policy. We observe that 5 participants found it *very reliable*, 19 of the participants found *somewhat reliable* while only 2 participants found *not at all reliable*.

CHAPTER 5

CONCLUSION

Decision-aid has always been interesting for complex decision making problem domains and we intend to present our approach of an agent Rae, a rational decision making agent designed to aid in complex decision making that can elicit decision problems of humans into a POMDP model.

We present our approach of preference elicitation for generating reward matrix of the problem domain. Preference elicitation has been used in designing models for understanding human understanding which is unique to every individual. We make use of this technique to understand user preferences of actions for states of their decision problem. We studied Analytical Hierarchical Process which is a technique to rank preferences on a scale of 1-9. Our approach is based on Analytical Hierarchical Process but we came up with our modified approach that assigns reward values for state-action pairs.

We present our approach of eliciting transitions, observations and generating policy based on the user's interaction with the agent. Our approach presents an interactive policy which the user can explore based on their observations. This policy represents a cyclic graph generated by SARSOP [5] as the policy and we represent it in an infinite tree that can be explored as action-observation choices. We accomplish this by coming with our own interpreter for dot files generated by SARSOP which converts dot files of POMDP policies into an interactive JavaScript based tree structures.

Our model is then evaluated by conducting a survey based on user interactions with our agent and finding correlations between the data. We present our hypotheses and provide correlations to evaluate our model. From this evaluation we find that our approach of designing a decision aid agent helps in elicitation of any problem into a POMDP model.

We are able to support Hypothesis 1 (page: 67) which tries to find whether Rae is an effective interactive decision aid that can satisfactorily elicit a decision problem as a POMDP.

We are able to support Hypothesis 2 (page: 73) which tries to find whether Rational policies obtained by solving acceptable POMDPs are utilitarian prescriptions.

We are able to support Hypothesis 3 (page: 74) which tries to find correlation between modal transitional probabilities and the judgment problem asked during the survey to demonstrate the external validity of the individual differences in probability assessment

We are able to support Hypothesis 4 (page: 77) which tries to find that participants who relate strongly to the problem, although they stand to benefit more from policy-based decision aids, may also have more contradictory thoughts about the topic and even though there is no direct measure of contradictory thoughts, relatability of the problem will correlate with policy satisfaction.

We are not able to support Hypothesis 5 (page: 79) which tries to find correlation between Modal transitional probabilities and policy satisfaction that reflects the response bias of participants associated with a correlation between individual differences in probability assessment and modal transition probabilities in policy satisfaction.

Our research will help in generating new real-world problem domains that users can elicit using our agent and this will contribute to research in artificial intelligence to have real-world problem domains to test features and developments in AI. It will also help as decision aid for human decision problems in many ways and in different fields since Rae is not limited to scope or subject of the domain problems and hence can be used for all purposes of complex decision making problems.

5.1 LIMITATIONS AND FUTURE WORK

This model is limited to single agent decision making process and accounts for solving decision problems involving single agent which means any effect from any other agent from the same environment is not accounted in the policy. Introduction of multi-agent systems in decision-aid introduces complexity and it is difficult to introduce multi-agent system in the current model.

Human usability study that we conducted was based on a very small number of participants and hence our analysis cannot be scaled to a larger public given low values of significance. Hence, this model needs to be scaled up and tested on larger set of participants in order to find more characteristics of human judgment, decision making and to design a better model based on this approach.

Our model needs to be scalable since the policy calculation is dependent on SARSOP which is executed through bash commands from Java API and calculate the policy based on values from the files such as ".POMDP", ".dot", etc., and hence it was difficult to scale the our model as there would have been wrong policy calculations if we had multiple threads of POMDP running on shared files.

Voice model for the agent is based on open-source APIs. Currently, the web-kit speech-to-text engine which is being developed through Mozilla as a W3 standard, is not yet good enough. For every speech-to-text error that is encountered during the NLP process, the error gets more weight for every new speech-to-text request and accuracy lowers gradually resulting in wrong responses from Rae. We implemented it as a feature but it is still a limitation as it needs to be worked upon for reducing the error affecting the NLP task.

Although, our model is designed to support as many states, actions and observations as per the problem domain, it may become difficult to use the model in case when there are more than 10 states, actions or observations as it may be time consuming to enter preferences, transitions and observations.

Future work can be based on introducing multi-agent model in decision-aid system. Improving scalability of the agent. This model is limited to an interactive policy tree which is a graph which may or may not have cycles, represented as an infinite tree of actions and observations. A better representation of the policy may help users. Making use of NLP and backtracking, elicitation of decision problems can be better represented with structures and graphs to explain the probability of actions, observations, states and their effect on the end goal state or the overall model.

BIBLIOGRAPHY

- [1] Russel Stuart J. and Norvig Peter. *Artificial Intelligence A Modern Approach*. Alan Ap, 1995
- [2] Braziunas, D. (2006). *Computational Approaches to Preference Elicitation*.
- [3] Chen L. and Pu P. *Survey of Preference Elicitation Methods*. EPFL Technical report, 2004.
- [4] Pack, L., Littman, M. L., & Cassandra, A. R. (1998). *Planning and acting in partially observable stochastic domains*. [https://doi.org/10.1016/S0004-3702\(98\)00023-X](https://doi.org/10.1016/S0004-3702(98)00023-X)
- [5] Kurniawati, H. (2008). *SARSOP : Efficient Point-Based POMDP Planning by Approximating Optimally Reachable Belief Spaces*. Robotics: Science and Systems 2008 Zurich,
- [6] Boutilier, C. (2002). *POMDP Formulation of Preference Elicitation Problems*. Proceedings of AAAI Conference on AI, 239–246.
- [7] Saaty, T. (1980). *The Analytic Hierarchy Process*. New York: McGraw-Hil. <https://doi.org/10.3414/ME10-01-0028>
- [8] Kahneman, B. Y. D., & Tversky, A. (1979). *Kahneman2013*, 47(2), 263–291. <https://doi.org/10.1016/B978-0-12-388437-4.00006-5>
- [9] Kreuter, F., Presser, S., & Tourangeau, R. (2008). *Social desirability bias in CATI, IVR, and web surveys: The effects of mode and question sensitivity*. Public Opinion Quarterly, 72(5), 847–865. <https://doi.org/10.1093/poq/nfn063>

- [10] Kim, K.-E. (2008). *Exploiting Symmetries in POMDPs for Point-Based Algorithms*. Proceedings of the 23rd AAAI Conference on Artificial Intelligence, 1043–1048. Retrieved from <https://www.aaai.org/Papers/AAAI/2008/AAAI08-165.pdf>
- [11] Ravindran, B., & Barto, A. G. (2001). *Symmetries and Model Minimization in Markov Decision Processes*.
- [12] Bocklisch, T., Faulkner, J., Pawlowski, N., & Nichol, A. (2017). *Rasa: Open Source Language Understanding and Dialogue Management*, (Nips), 1–9. Retrieved from <http://arxiv.org/abs/1712.05181>
- [13] Adorf, J. (2013). *Web Speech API*. KTH Royal Institute of Technology, Stockholm, 1–11. Retrieved from <http://home.in.tum.de/~adorf/pub/web-speech-api.pdf>
- [14] Furnham, A. (1986). *Response bias, social desirability and dissimulation*. *Personality and Individual Differences*, 7(3), 385–400. [https://doi.org/10.1016/0191-8869\(86\)90014-0](https://doi.org/10.1016/0191-8869(86)90014-0)
- [15] Taha, T., Miro, J. V., & Dissanayake, G. (2011). *A POMDP framework for modelling human interaction with assistive robots*. Proceedings - IEEE International Conference on Robotics and Automation, (May), 544–549. <https://doi.org/10.1109/ICRA.2011.5980323>
- [16] Huang, Dongling, Luo, L. (2016). *Consumer Preference Elicitation of Complex Products Consumer Preference Elicitation of Complex Products Using Fuzzy Support Vector Machine Active Learning*. *Marketing Science*, (March 2017).
- [17] PHILLIPS-WREN, G. (2012). *Ai Tools in Decision Making Support Systems: a Review*. *International Journal on Artificial Intelligence Tools*, 21(02), 1240005. <https://doi.org/10.1142/s0218213012400052>

- [18] American Economic Association *Designing Economic Agents that Act like Human Agents : A Behavioral Approach to Bounded Rationality* Author (s): W . Brian Arthur Source : The American Economic Review , Vol . 81 , No . 2 , Papers and Proceedings of the Hund. (2018), 81(2).
- [19] Briot, J.-P., Sordoni, A., Vasconcelos, E., Sebba Patto, V., Adamatti, D., Irving, M., & Lucena, C. (2009). *Design of an Artificial Decision Maker for a Human-based Social Simulation – Experience of the SimParc Project*. Cargese '09 Seminar, 1–16.
- [20] Vasconcelos, E., Lucena, C., Melo, G., Irving, M., Briot, J. P., Sebba, V., & Sordoni, A. (2009). *A serious game for exploring and training in participatory management of national parks for biodiversity conservation: Design and experience*. SBGAMES2009 - 8th Brazilian Symposium on Games and Digital Entertainment, 93–100. <https://doi.org/10.1109/SBGAMES.2009.19>
- [21] Boman, M., Davidsson, P., & Younes, H. L. (1999). *Artificial Decision Making Under Uncertainty in Intelligent Buildings*. Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence, 65–70. Retrieved from <http://dl.acm.org/citation.cfm?id=2073796.2073804>
- [22] C.C., B., & K., H. (2013). *Artificial intelligence framework for simulating clinical decision-making: A Markov decision process approach*. Artificial Intelligence in Medicine, 57(1), 9–19. <https://doi.org/10.1016/j.artmed.2012.12.003>

- [23] Barnett, G. O., Cimino, J. J., Hupp, J. A., & Hoffer, E. P. (1987). *DXplain: An Evolving Diagnostic Decision-Support System*. JAMA: The Journal of the American Medical Association. <https://doi.org/10.1001/jama.258.1.67>
- [24] Bilimoria, K. Y., Liu, Y., Paruch, J. L., Zhou, L., Kmieciak, T. E., Ko, C. Y., & Cohen, M. E. (2013). *Development and evaluation of the universal ACS NSQIP surgical risk calculator: A decision aid and informed consent tool for patients and surgeons*. Journal of the American College of Surgeons, 217(5), 833-842.e3. <https://doi.org/10.1016/j.jamcollsurg.2013.07.385>
- [25] Knops, A. M., Legemate, D. A., Goossens, A., Bossuyt, P. M. M., & Ubbink, D. T. (2013). *Decision aids for patients facing a surgical treatment decision: A systematic review and meta-analysis*. Annals of Surgery, 257(5), 860–866. <https://doi.org/10.1097/SLA.0b013e3182864fd6>
- [26] Ubbink, D. T., Knops, A. M., Molenaar, S., & Goossens, A. (2008). *Design and development of a decision aid to enhance shared decision making by patients with an asymptomatic abdominal aortic aneurysm*. Patient Preference and Adherence, 2, 315–322. <https://doi.org/10.2147/PPA.S4302>

APPENDIX A

SURVEY 1

Question 1: Which decision problem did you choose?

Options: (Single choice, since we had only one problem domain to test the experiment.)

1. *Change of Major Decision Problem*

Question 2: Which of the following is a state?

Options:

1. *Tiger behind the left door (TL)*
2. *Listen to Growl (L)*
3. *Growl from the right (GR)*

Question 3: Which of the following is an action/decision choice?

Options:

1. *Tiger behind the left door (TL)*
2. *Listen to Growl (L)*
3. *Growl from the right (GR)*

Question 4: Which of the following is an observation?

Options:

1. *Tiger on the left (TL)*
2. *Listen to Growl (L)*
3. *Growl from the right (GR)*

Question 5: Which of the following is a transition?

Options:

1. *"Tiger on the left" To "Tiger on the left"*
2. *"Tiger on the left" To "Growl from the right"*
3. *"Tiger on the left" To "Listen"*
4. *"Open Left Door" To "Open right door"*

Question 6: Which of the following is/are a 'correct order for preference'?

Options: (Multiple choices correct)

1. *For state Tiger behind the right door (TR): Open right door(OR) , Open left door (OL), Listen (L)*
2. *For state Tiger behind the left door(TL): Open right door(OR), Open left door(OL), Listen (L)*
3. *For state Tiger behind the right door(TR): Open left door (OL), Listen (L)*

Question 7: How much did you relate the Change of Major domain with your own decision problem?

Options:

1. *Not at all relatable*
2. *Somewhat relatable*
3. *Very relatable*

APPENDIX B

SURVEY 2

Question 1: On a scale of 1 to 5, how satisfied are you with the suggested policy of the decision problem?

Options: 1 to 5, '1' : *Very unsatisfied*, '5' : *Very satisfied*

Question 2: Which part of the policy you are not satisfied with? For this question, you need to provide a brief answer specifying what part of policy you think should have been different. You may specify a different policy as : *action1* – > *observation1* – > *action2* – > *observation2*, etc., although not necessary.

Options: (Descriptive answer expected from the participant)

Question 3: Transition probabilities that you entered should have been different?

Options: *Yes, No, Maybe*

Question 4: Imagine a situation where you are walking down the street without an umbrella and if you see a patch of dark-grey clouds on one side and clear sky with Sun on the other side, what do you think would be the chance that it will rain?

Options: Choose only one option from: 49%, 56%, 43%, 65%, 27%, 73%, 18%, 89%, 91%, 11%, 98%, 1%

Options were randomly shuffled for every participant

Question 5: Do you think the order of states you entered as preferences for each action-observation should have been different?

Options: *Yes, No, Maybe*

Question 6: Enter your preferred order of states you think would have given a better policy. For this question, you need to provide a brief answer specifying the order of states for one specific/all action-observations that you think should have been different. You may enter it as as : action-observation: state1, state2, etc., by replacing 'action', 'observation', 'state1',... with labels of your decision problem.

Options: (Descriptive answer expected from the participant)

Question 7: Do you think the order of actions you entered as preference for each state should have been different?

Options: *Yes, No, Maybe*

Question 8: Enter your preferred order of actions for every state that you think would have given a better policy

For this question, you need to provide a brief answer specifying the order of actions for one specific/ all states that you think should have been different. You may enter it as as : state1: action1, action2, etc., by replacing 'action1', 'state1',... with your labels of the decision problem.

Options: (Descriptive answer expected from the participant)

Question 9: How helpful were Rae's replies?

Options: 1 to 5, '1' : *Not helpful*, '5' : *Very helpful*

Question 10: How satisfied are you with the working of Rae?

Options: 1 to 5, '1' : *Least satisfied*, '5' : *Very satisfied*

Question 11: Do you want to suggest anything for improving Rae? What went wrong or what could have been better?

Options: (Descriptive answer expected from the participant)