2012

An Approach for Contextual Control in Dialogue Management with Belief State Trend Analysis and Prediction

Rajaprabhu Dhanapal
University of Windsor

Follow this and additional works at: https://scholar.uwindsor.ca/etd

An Approach for Contextual Control in Dialogue Management with Belief State Trend
Analysis and Prediction

By

Rajaprabhu Dhanapal

A Thesis
Submitted to the Faculty of Graduate Studies
Through Computer Science
In Partial Fulfillment of the Requirements for
The Degree of Master of Science at the
University of Windsor

Windsor, Ontario, Canada

2012

An Approach for Contextual Control in Dialogue Management with Belief State Trend
Analysis and Prediction

By

Rajaprabhu Dhanapal

APPROVED BY:

_____
Dr. Patti Fritz
Department of Psychology

_____
Dr. Alioune Ngom
School of Computer Science

_____
Dr. Xiaobu Yuan, Advisor
School of Computer Science

_____
Dr. Luis Rueda, Chair of Defence
School of Computer Science

DECLARATION OF ORIGINALITY

I hereby certify that I am the sole author of this thesis and that no part of this thesis has been published or submitted for publication.

I certify that, to the best of my knowledge, my thesis does not infringe upon anyone's copyright nor violate any proprietary rights and that any ideas, techniques, quotations, or any other material from the work of other people included in my thesis, published or otherwise, are fully acknowledged in accordance with the standard referencing practices. Furthermore, to the extent that I have included copyrighted material that surpasses the bounds of fair dealing within the meaning of the Canada Copyright Act, I certify that I have obtained a written permission from the copyright owner(s) to include such material(s) in my thesis and have included copies of such copyright clearances to my appendix.

I declare that this is a true copy of my thesis, including any final revisions, as approved by my thesis committee and the Graduate Studies office, and that this thesis has not been submitted for a higher degree to any other University or Institution.

ABSTRACT

This thesis applies the theory of naturalistic decision making (NDM) in human physcology model for the study of dialogue management system in major approaches from the classical approach based upon finite state machine to most recent approach using partially observable markov decision process (POMDP). While most of the approaches use various techniques to estimate system state, POMDP-based system uses the belief state to make decisions. In addition to the state estimation POMDP provides a mechanism to model the uncertainty and allows error-recovery. However, applying Markovian over the belief-state space in the current POMDP models cause significant loss of valuable information in the dialogue history, leading to untruthful management of user's intention. Also there is a need of adequate interaction with users according to their level of knowledge. To improve the performance of POMDP-based dialogue management, this thesis proposes an enabling method to allow dynamic control of dialogue management. There are three contributions made in order to achieve the dynamism which are as follows: Introduce historical belief information into the POMDP model, analyzing its trend and predicting the user belief states with history information and finally using this derived information to control the system based on the user intention by switching between contextual control modes. Theoretical derivations of proposed work and experiments with simulation provide evidence on dynamic dialogue control of the agent to improve the human-computer interaction using the proposed algorithm.

DEDICATION

To my dear parents, my brother and my fiancée

ACKNOWLEDGEMENTS

First, I would like to thank and express my sincere gratitude to my supervisor Dr. Xiaobu Yuan, for his support and encouragement with his valuable hints and stimulating suggestions to proceed with this thesis. Without his support and guidance, this work would be impossible.

Secondly, I would like to take this opportunity to thank Dr. Alioune Ngom and Dr. Patti Fritz, for being in the thesis committee and spending their valuable time in providing me with encouraging feedback.

Finally I would like to thank my friend Mr. Sivaraman Sriram for his encouragement and moral support.

TABLE OF CONTENTS

CHAPTER

1. Introduction

2. Preliminary

3. Literature Survey

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

Introduction

Dialogue is a conversation between two or more agents, be they human or machine. Research on dialogue usually follows two main directions: human-human dialogue and human-computer dialogue. The later is involved in a dialogue system, a computer program that communicates with a human user in a natural way. Previous research work has been focusing on spoken dialogue systems, which are defined as computer systems that human interact on a turn-by-turn basic and in which spoken natural language interface plays an important part in the communication recently, it has been extended to multimodal dialogue systems, which are dialogue systems that process two or more combined user input modes - such as speech, pen, touch, manual gestures, gaze, and head and body movements - in a coordinated manner with multimedia system output. Both spoken dialogue system and multimodal dialogue system need a central management module called the dialogue manager [1]. The dialogue manager (DM) is the program which coordinates the activity of several subcomponents in a dialogue system and its main goal is to maintain a representation of the current state of the ongoing dialogue. This thesis report describes the sequential decision making and control problems in dynamic environments with incomplete and uncertain information using the Partially Observable Markov Decision Process (POMDP) frame work with trend information available. Designing agents that can act under uncertainty is mostly done by modeling the environment as a Partially Observable Markov Decision Process. In POMDPs, an agent

1

interacts with a stochastic environment at discrete time steps. The agent takes actions, and as a result, receives observations and rewards. The agent then has to find a way of choosing actions, or policy, which maximizes the total reward received over time. Most POMDP planning methods try to construct a Markovian state signal using a model of the environment and the history of actions and observations experienced by the agent. This signal is called a belief state. Planning methods then use reward information in order to associate an (optimal) action to each belief state.

The information related to the belief states of the past by different user will be learned by the agent. This history helps the agent to predict the forward trend of the belief state which intern understand the user's knowledge about the domain and the intention of participation. The multimodal dialogue management system is studied with naturalistic decision making which is theory of human psychology for best decision making. The trend predicted with the history information will reduce the uncertainty of the state of the agent and thereby understanding the user. Belief states can identify more precisely the hidden state of the system. If the hidden state were known with better precision, the action choices of the agent could be better as well. The changes in the trend of the belief state in the belief space with the history information space will help in the process of effective planning for the construction of a real truthful, relevant, clear, informative dialogue management.

1.1 The Problem Statement

Various approaches of dialogue management have been proposed in the last twenty years, including the classical approach based upon finite state machine and the current approach based upon the popular POMDP model. Finite state machine based approach is only suitable to the well structured task and is lack of flexibility. Frame based approach uses a frame to record the information and is more flexible than finite state machine based approach. Bayes network and Markov Decision Process based approach are probabilistic which can solve some uncertainties to some degree but still have drawbacks such as defects in solving observation uncertainties. Although POMDP based approach is the current popular approach, it still has its own problems to be taken care of. Despite its known problem of scalability, the POMDP-based approach demonstrates undeniable advantages in the handling of input uncertainty over other approaches. However, applying the Markovian over the belief-state space in the current POMDP models causes significant loss of valuable information in dialogue history, leading to untruthful recognition of user intention. In other perspective, the POMDP-based approach only models the user and maintains the knowledge at the control level. However the POMDP model does not analyzes the user intention and level of domain knowledge and treats every user in the same way no matter how much domain knowledge they have and what is their intention.

1.2 Contribution

There are three contributions made in this thesis. First, belief history information is introduced into the POMDP based dialogue management system. Secondly, the history information is used by the dialogue manager and analyzes the trend of belief states to find the rate of change in trend and also predicts the next belief state using machine learning techniques. Third, based on the change in belief trend allows the agent to handle different user switching between contextual control modes in different stages to improve the human - computer interaction.

We have modeled a dialogue manager for a Software requirement customization and modified the traditional POMDP dialog manager. The modified architecture with the trend analyzed values in belief history will help the agent to know well about the user intention and the knowledge in the domain thereby making decisions in an improved way. The experiments under 5 scenarios are conducted to evaluate the proposed method. The results of the experiments prove that the concept and they demonstrate that the proposed method achieves the expected results.

1.3 Organization of the thesis

In the remaining of part this thesis, chapter 2 provides the preliminary information on different components of dialogue management system. Chapter 3 provides a literature review about the major approaches of dialogue management. Chapter 4 discusses the previous work on history information space and trend and then conducts an analysis of all the major approaches but the POMDP-based approach. Chapter 5 discusses the proposed method of belief state trend analysis and prediction to analyze the change in user intention thereby switching the contextual control mode to handle different user groups. Three types of experiments are presented in chapter 6, whose results show that the new approach is more accurate in the recognition of user intention, thus making agents more attractive and useful when providing services. Finally, chapter 7 ends with conclusions and points out directions for future work.

1.4 Motivation

In the last decades, there has been a tremendous development in the field of virtual reality, particularly the research related to dialogue management system. The dialogue management system is a setup made to help the human by an agent. In short, agent is a replacement of human to help other human. The system has to behave and produce result similar to the way as how a human does. The researches so far done are to improve the algorithm used in the agent to identify the state which is uncertain and reduces the ambiguity to the maximum. Use of POMDP algorithm will help the agent to reduce the ambiguity to a certain level and still it loses the information related to the domain which is solved by adding a domain constraint module given by Libian. The study of Naturalistic decision making theory states clearly that pattern analysis and decision making in human brain. Applying the NDM theory in the dialogue management system will help the agent to classify the user type and the prediction of the user's next possible states with the past experience by the agent in the dialogue history. This will reduce the ambiguity of the agent state in the dynamic environment thus the decision making happens more efficient. This motivates me to conduct a research on this particular technology and solve the challenges associated with POMDP based dialog manager by introducing belief trend and based on four-mode concept.

CHAPTER 2

Preliminary

In this chapter, the definition of the dialogue system and dialogue manager will be introduced and also the basic issues in the dialogue system and dialogue manager will be discussed.

2.1 Spoken Dialogue System

Spoken dialog systems (SDS) help people achieve a task using spoken language. For example, a person might use an SDS to buy a train ticket over the phone, to direct a robot to clean a bedroom, or to control a music player in an automobile. Building SDS is a challenging engineering problem in large part because automatic speech recognition (ASR) and understanding technology are error-prone. More specifically, speech recognition accuracy is relatively good for constrained speech limited to, for example, digits, place-names, or short commands, but accuracy degrades rapidly as the domain language becomes less constrained. Furthermore, as spoken dialog systems become more complex, not only do the demands on the speech recognition and understanding components increase, but also user behavior becomes less predictable. Thus, as task complexity increases, overall there is a rapid increase in uncertainty, and principled methods of dealing with this uncertainty are needed in order to make progress in this research area [2].

The goal of dialogue management in a spoken dialogue system is to take actions based on observations and inferred beliefs. Dialogue management plays a crucial role in the

overall performance of the system since speech recognition is often quite poor, due to noisy or unexpected input. With robust dialogue management, the system can still take actions that maintain the task at hand. Unfortunately, coming up with a suitable set of dialogue management strategies is no easy task. Traditional methods typically involve authoring and tuning complicated hand-crafted rules that require considerable deployment time and cost. Statistical methods, on the other hand, hold the promise of robust performance from models that can be trained on data and optimized, so long as the data is representative of what the dialogue system can expect to encounter during deployment [2].

[3] Identified that the main purpose of a spoken dialogue system is to provide an interface between a human user and a machine usually computer-based application such as a database or expert system. Also Mctear identified that the main tasks of the dialogue system include processing the user's input and recovering from the errors. Based on this, Mactear categorized different dialogue strategies into three types: finite state or graph based approach, frame based approach and agent based approach. Later in 2006, [4] identified that current spoken dialogue system had been extended to multimodal dialogue system, which means that the dialogue systems can process two or more combined user input modes such as speech, pen, touch, manual gestures, gaze, and head and body movements, etc in a coordinated manner with multimedia system output. Bui meanwhile identified that the central module of the spoken dialogue system and multimodal system is the dialogue manager (DM). The function of the DM is to coordinate the activity of corresponding subcomponents in a dialogue system and its main goal is to maintain a representation of the current state of the ongoing dialogue.

Spoken dialogue systems can be classified into three main types, according to the

Methods used to control the dialogue with the user:

(1) Finite state- (or graph-) based systems;

(2) frame-based systems; and

(3) agent-based systems.

The type of dialogue control strategy used has a bearing on how the system accomplishes

two of its main tasks: processing the user's input and recovering from errors [5]. It

discussed each component in the spoken dialogue system and their functionalities.

The involved components explained in [5] are as follows:

• *Speech recognition* which converts an input speech utterance consisting of a sequence

of acoustic-phonetic parameters into a string of words.

• *Language understanding*, the component is analyzing a string of words with the aim of

producing a meaning representation for the recognized utterance. The produced meaning

representation can be used by the following dialogue management component.

• Dialogue Management is the control component of the interaction between the system

and the user. It is also responsible for coordinating with other components of the system.

• Communication with external system is, for example, a database system, expert system,

or other computer application.

• Response generation which is the specification of the message to be output by the

system.

• Speech output is the component to apply text-to-speech synthesis or pre-recorded

speech techniques to output the system's message. In Bui's research work, he described

the multimodal dialogue system containing components of Input, Fusion, Dialogue Manager (DM), Knowledge Sources, Fission, and Output. Inputs of a multimodal dialogue system can be any subset of the modalities. The following fusion component receives the extracted information from the input modalities and passes the processed information usually a semantic structure to a dialogue manager. Dialogue manager takes this semantic structure as the observation to generate appropriate response. By coordinating with other component, DM sends its output to the fission component. The information received by the fission component along with output component will also be processed to generate human natural language responses to the human user. The Fig 2.1 illustrates the overall multimodal dialogue system structure and relations among all the components.

Dialogue manager is the most important component in the (multimodal) dialogue system. The main functions of the DM include, coordinating with other components, identifying the intention of the user's intention and deciding what to respond to the user at what time steps. In [7], the main tasks of dialogue manager are identified as following:

• Updating the dialogue context on the basis of interpreted communication

• Providing context-dependent expectations for interpretation of observed signals as communicative behavior.

• Interfacing with task/domain processing (e.g., database, planner, execution module, other back-end system), to coordinate dialogue and non-dialogue behaviors and reasoning.
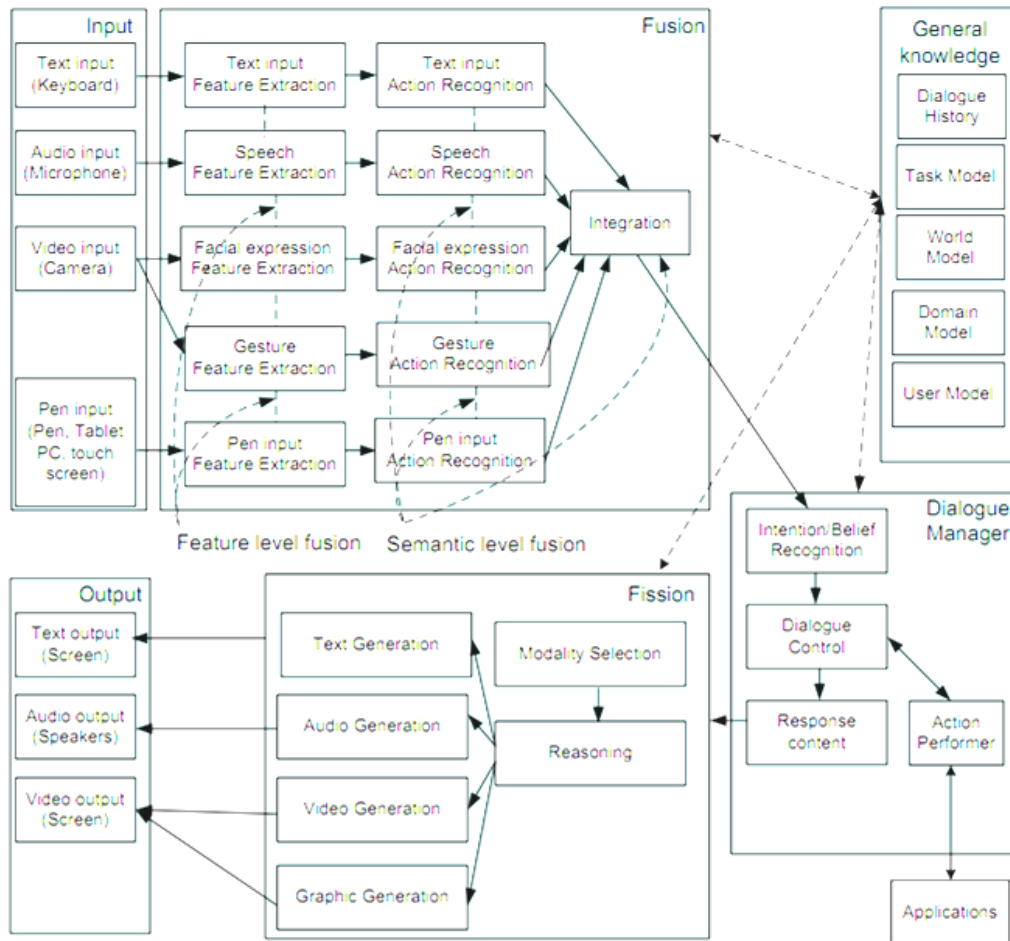
Figure 2.1: Overall Multimodal Dialogue System Structure

2.2 Dialogue Manager

Dialogue Manager is the core module of the system. The main tasks of DM are [7]:

- Updating the dialogue context on the basis of interpreted communication.

- Providing context-dependent expectations for interpretation of observed signals as communicative behavior.

- Interfacing with task/domain processing (e.g., database, planner, execution module, other back-end system), to coordinate dialogue and non-dialogue behavior and reasoning.

- Deciding what content to express next and when to express it.

The term "dialogue context" can be viewed as the totality of conditions that may influence the understanding and the generation of communicative behavior [8]. This definition is quite vague, and Bunt restricts to "local" aspect of the dialogue context (also called local context) which can be changed through communication. Local context factors can be grouped into five categories of conceptually different information dimensions: Linguistic, cognitive, Physical, semantic, and social as shortly described in table 1. More detail about these contexts is described in [8].

Linguistic context	Surrounding linguistic material, `raw' as well as analyzed
Semantic context	Semantic context state of the underlying task; facts in the task domain.
Cognitive context	Participants' states of processing and models of each other's states.
Physical and Perceptual	Availability of communicative and perceptual channels; partners' presence and attention.
Social	Context communicative rights, obligations and constraints of each participant.

The dialogue manager may draw on a number of knowledge sources, which are sometimes referred to collectively as the dialogue model. A dialogue model might include the following types of knowledge relevant to dialogue management:

A dialogue history: A record of the dialogue so far in terms of the propositions that have been discussed and the entities that have been mentioned. This representation provides a basis for conceptual coherence and for the resolution of anaphora and ellipsis.

A task record: A representation of the information to be gathered in the dialogue. This record, often referred to as a form, template, or status graph, is used to determine what information has not yet been acquired. This record can also be used as a task memory [6] for cases where a user wishes to change the values of some parameters, such as an earlier departure time, but does not need to repeat the whole dialogue to provide the other values that remain unchanged.

A world knowledge model: This model contains general background information that supports any commonsense reasoning required by the system, for example, that Christmas day is December 25. A domain model: A model with specific information about the domain in question, for example, flight information. A generic model of conversational competence: This includes knowledge of the principles of conversational turn-taking and discourse obligations.

A user model: This model may contain relatively stable information about the user that may be relevant to the dialogue such as the user's age, gender, and preferences—as well as information that changes over the course of the dialogue, such as the user's goals, beliefs, and intentions.

After that the knowledge sources probably used in the DM are discussed, it comes to the problem of modeling based on the knowledge sources. Therefore, the distinction between Dialogue modeling and dialogue management modeling [9] should be also explained here. The goal of dialogue modeling is to develop general theories of dialogues such as task oriented and to investigate the similarities between the courses of the dialogues. Dialogue modeling is to provide dialogue management with theoretical support. While, the goal of dialogue management modeling is to combine dialogue model with task model under particular domain to design algorithms which support a machine's decision making in a dialogue, or it can be said that it takes the viewpoint of a dialogue system designer. Dialogue manager executes based on the dialogue policy. Dialogue policy is "What the system should do next to respond to the users", which maps from a set of states in the state space to a set of actions. Usually, the actions of a dialogue manager can be divided into five types including: greeting, submitting, initiative, repeating and confirmation. And at each stage, the action taken by the system can receive different results or it can receive various rewards or costs. In some stage, such as the first round of the dialogue, the greeting should always be the most appropriate action taken by the system. While during the whole dialogue, the action of initiative, repeating and confirmation are always not clear. At each round, different actions correspond to different rewards. In the past efforts made on the dialogue management, various approaches have been proposed to resolve the dialogue policy generation problem.

CHAPTER 3

Literature Survey

In this Chapter, Dialogue management approaches and their classification are discussed in the first section. In the second and third section Information state, Probabilistic are discussed. In the third section literature review about prediction of user metal states in spoken dialogue system is done. The last three sections discuss about decision making processes and techniques involved

3.1   Dialogue Management Approaches Classification

There has been an active research conducted in the past two decades towards dialogue management. There are two aspects to dialogue control one is the extent to which one of the agents maintains the initiative in the dialogue and the ways in which the flow of the dialogue is managed. Dialogue control may be system-led, user-led, or mixed-initiative. In a system-led dialogue the system asks a sequence of questions to elicit the required parameters of the task from the user. In a user-led dialogue the user controls the dialogue and asks the system questions in order to obtain information. The control strategy of a dialogue system may use finite states, frame slots, autonomous agents [3] or Bayesian networks and decision graphs approach [10]. Some dialogue strategies may be generated by the plan-based approach [13] and this is based on the view that humans communicate to achieve goals, collaborative agent-based approach [11] and this evaluates the dialogues

as collaboration between two intelligent agents to achieve mutual understanding of the dialogue or theorem proving approach [12]. Different dialogue management approaches have been classified into several categories by the researchers. According to [14] task model and dialogue model are used; approaches to dialogue management can be classified into four categories in Table 3.1.

		Task Model	
		implicit	explicit
Dialogue	implicit	DITI	DITE
Model	explicit	DETI	DETE

Table 3.1 Classifying dialogue management models

[4]. For example, the frame based approach is usually used by combining with probabilistic method. Basically, there are main five types of dialogue management including finite state machine, frame base, Bayes network, Markov Decision Process based and POMDP based approach based upon the recent development of information state and probabilistic methods.

3.1.1 Information State Approaches

Dialogue systems can provide a great test bed for theories of dialogue, since they can straightforwardly manifest behavior of an implemented theory as the dialogue progresses; however, this is true only in so far as the system incorporates an accurate representation of the theory. To help in this regard, [15] present a method of specifying a dialogue theory that makes it straightforward to implement, and, as described in the following sections, tools to help implement a dialogue theory specified along these lines.

To overcome the limitations of previous approaches, Information state-based approach is a dialogue theory with five different components; each has its own functionality [15]. An

informational component is to track the intentional structure and user models. Formal representation is for the discourse representation structures, modal operators within a logic etc. A set of update rules for updating the information state and a set of dialogue moves to trigger the update of information state. An update strategy is to decide which rule to apply. The general idea of this approach is to develop the multi-layer dialogue model. In this model, each level contains an information state representing current status of the layer. Trindikit toolkit is developed based on this approach followed by GodiS [16] and EDIS [17]. Several other applications of this approach include MATCH system for multimodal city help [18], Virtual Music Center [19], etc.

Frame based approach can realize the mix imitative dialogue and tolerant redundant information brought by the users. The sequence of the questions or the information to be gathered is not pre-determined, which is based on the current context to generate next question to ask. However, Mctear in 2002 summarized that the next step only based on the current context is not enough. More complicated domain in which the state of the world is dynamic or the knowledge level of the user is varied can not apply for the frame based approach.

3.1.2 Probabilistic Approaches

The improvement in the performance of dialogue management has been concentrated by serveral research groups recently. This approach can be considered as the extension of the information state approaches [20]. The techniques include Markov Decision Process (MDP) or Partially Observable Markov Decision Process (POMDP). The basic idea is to overcome the limitations of Multilayer dialogue model and to provide dynamically changing actions and dialogue strategy based on rewards of the current state. The

dialogue model is designed to use optimal strategy using some reinforcement learning. The system actions are modeled to system's question and answers, the rewards are pre-set by the system to rate the dialogue or it is provided to the user to rate the system at the end of each dialogue [22]. In [2], dynamic programming, Q-learning or sampling-based reinforcement learning is used to optimize the dialogue cost function. Inductive logic programming is to extract rules from the result of reinforcement learning. Apart from the MDP and POMDP techniques, Bayesian Networks are also used to recognize the dialogue acts or to control the dialogue strategy.

Wai et al. [22] proposed to use of Belief networks (BN) for mixed-initiative dialogue modeling. They applied their approach into the CU FOREX system which is a bilingual hotline for real time foreign exchange inquiries. The author adopted Belief networks in mixed-initiative dialog modeling involving the following two processes: inferring the informational goal of a user's query and verifying the input query against domain-specific constraints. In the process of goal identification, a BN is trained for each domain-specific Informational goal and then it is used to make a binary decision based on the concepts present in the input query. With the decisions across all BNs combined, the output goal can be identified regarding the input query. Followed by the backward inference process, the validity of the input query will be verified. The system responses can be generated based on the result of the spurious and missing concepts detection process. In 2003, Wai et al. migrated their dialog model from the simple foreign exchange domain to air travel information service domain. In this work, they described the scalability and portability of a Belief Network based mixed initiative dialog model across application domains.

Paek and Horvitz [23] proposed using Decision Networks as the dialogue model to manage a hidden sub dialog. Paek and Horvitz stated that the problem that when the dialogue system attempts to solicit information from the user, it may have to engage in a hidden sub dialog or error handling in a particular state. They considered that hidden sub dialogs generally centers on illocutionary repairs including asking for repeating or conforming, etc. It is described that there are three advantages by applying the decision network in dialog management: first the propagation of uncertainties over time to assist recognition, second the ability to leverage key contextual dependencies, such as the acoustic environment, and the consideration of the stakes involved in taking real-world actions. Williams et al. in [24] stated that this approach selects the action only based on the immediate maximum expected utility and in this scenario this proposal can be treated as a POMDP that greedily selects the actions.

Other important groups of researchers have delved their efforts into probabilistic techniques such as (fully observable) Markov Decision Process (MDP) or a Partially Observable Markov Decision Process (POMDP) as the dialogue model to resolve the action outcome and observation uncertainties existed in the human-computer interaction process. [25] and [26] all cast the dialogue management problem as the MDP problem with the assumption that a good dialogue strategy is minimizing an objective function that reflects the costs of all the important dialogue dimensions. Levin et al. stated that allowing a user to change the course of dialogue or to change request during dialogue in a mixed-initiative system could result in a branching factor and make the tree prohibitively large. Therefore, they adopt the Markov Decision process approach. The operation of the dialogue manager based on the Markov Decision Process described by Levin et al .

Initialization: start from initial state

Iterate until done (final state is reached)

Next Action: Choose and perform next action

Get new input

Next State: Update state with new input

Roy et al. was the first group to treat dialogue management as a problem of partially observable Markov decision process [27]. They noticed that the MDP approach cannot handle noise and ambiguity in speech utterances. They used POMDP models to generate dialogue strategy and used, rather than estimated system state, belief state to represent user intention. They conducted experiments and claimed that the POMDP-based dialogue system made fewer mistakes than MDP-based dialogue system. With increased errors in automated speech recognition in real-life situations, the advantage of uncertainty handling is obvious.

[28] Address several challenges for applying statistical dialog managers based on Partially Observable Markov Models to real world problems: to deal with large numbers of concepts, [28] use individual POMDP policies for each concept. To control the use of the concept policies, the dialog manager uses explicit task structures. The POMDP policies model the confusability of concepts at the value level. In contrast to previous work, [28] use explicit confusability statistics including confidence scores based on real world data in the POMDP models. Since data sparseness becomes a key issue for estimating these probabilities, [28] introduce a form of smoothing the observation probabilities that maintains the overall concept error rate.

3.1.2.1 POMDP Models

POMDP Model can be divided into two types including flat POMDP which the state only contains user belief component and factored POMDP which extends the state of flat POMDP to integrate user action and dialogue state. In the following of this section, both Flat POMDP model and factored POMDP model will be reviewed.

3.1.2.1.1 Flat POMDP Model

In a POMDP system, the state of the system is not observable and therefore unknown to the decision process. Action selection depends on the decision made over belief state, denoted by b. Formally, a POMDP is defined as a tuple { S, Am, T, O, Z, R} , where S is a set of states, Am is a set of actions the system may take, T is the transition model that defines transition probability, O is a set of observations from user's actions, Z is the observation model that defines the observation probability, and R defines the immediate expected real valued reward r(s, am). And also b is the agent's belief state and π is the agent's policy to select action. With the assumption that the state and O are both discrete and based on the above notation and definition, the operation process of POMDP can be described as following:

POMDP system carries out two tasks. The first task is to compute or update belief state, and the second is to find an optimal policy [29]. With the latest belief state and the offline computed optimal policy, the agent can perform appropriate action checking to select action to take. For the first task, the belief state is updated at each time step based upon the Bayes filter algorithm. Bayes filter algorithm is under the Markov assumption. The Markov decision process makes assumption that the action of nature only depend on the current state and action as opposed to the state or action histories. In [30], the Markov

21

Assumption and Bayes Filters framework were describes as following. Markov assumption is with underlying assumption that the world is static; the noise is independent and perfect models no approximation error. The Markov assumption allows the recursive Bayesian updating to be used to efficiently combine evidence. The Markov Assumption illustrated in Dynamic Bayes network is shown in Fig 5.1.
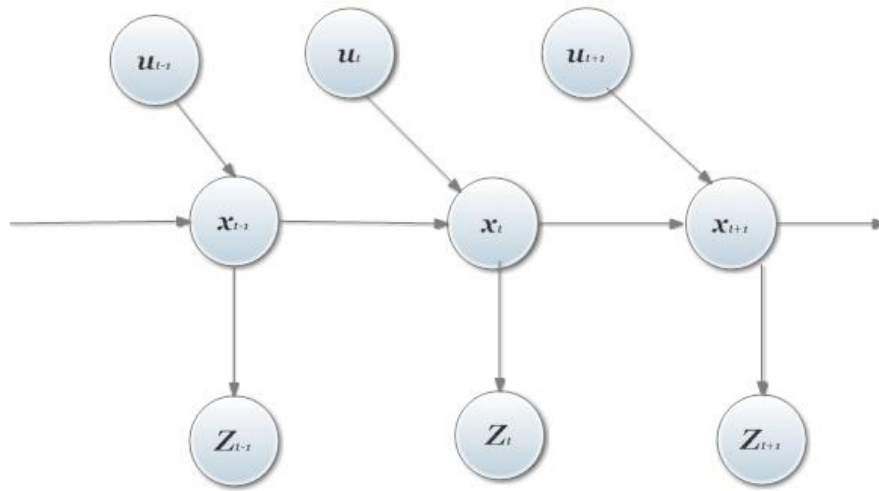

Figure 3.1: Markov Assumption illustrated in Dynamic Bayes Network

The expected observation probability depends on local information. Here the local information means that distribution depends only on information obtained at the current stage. And the posterior probability over state depends on the previous state and newly taken action.

Based on the above Markov assumption, Bayes filter is the probabilistic method to estimate state in dynamic environment. The estimation of state computation process is shown in Fig 5.2. Thus, the computation of belief state uses the following equation, where a is the normalizing constant, $P(O_{t+1}|S_{t+1}, a_t)$ is the observation model or named sensor model and $P(S_{t+1}|S_t, a_t)$ is the action model or named transition model.

$$b_{t+1}(S_{t+1}) = \alpha P(O_{t+1}|S_{t+1}, a_t) \sum P(S_{t+1}|S_t, a_t)b_t(S_t)$$

Equation 1 the estimation of state computation process

$$Bel(s_t) = P(s_t \mid a_1, o_1 ...., a_t, o_t)$$

$$= \alpha \ P(o_t \mid s_t, a_1, o_1, ..., a_t) \ P(s_t \mid a_1, o_1, ..., a_t)$$

$$= \alpha P(o_t \mid s_t) \ P(s_t \mid u_1, o_1, ..., u_t)$$

$$= \alpha \ P(o_t \mid s_t) \int P(s_t \mid a_1, o_1, ..., a_t, s_{t-1}) P(s_{t-1} \mid a_1, o_1, ..., a_t) \, ds_{t-1}$$

$$= \alpha P(o_t \mid s_t) \int P(s_t \mid a_t, s_{t-1}) \ P(s_{t-1} \mid a_1, o_1, ..., a_t) \, ds_{t-1}$$

$$= \alpha P(o_t \mid s_t) \int P(s_t \mid a_t, s_{t-1}) P(s_{t-1} \mid a_1, o_1, ..., o_{t-1}) \, ds_{t-1}$$

$$= \alpha P(o_t \mid s_t) \int P(s_t \mid a_t, s_{t-1}) \ Bel(s_{t-1}) \, ds_{t-1}$$

Equation 2 Estimation of State using Bayes Filter

For the current belief state, Eq. 2 constitutes the flat POMDP model that selects an optimal policy as the maximum of all the expected value function $V^\pi(b)$ with a discounted future reward starting from b for a policy π.

$$\pi^* = argmax_\pi E[V^\pi(b)]$$

Equation 3: Optimal policy Computation

3.1.2.1.2 Factored POMDP

In 2005, William et al. casted the spoken dialogue system as a factored POMDP to use this model as general framework for existing POMDP dialog manager. In this model, the POMDP state variable s ϵ S into three components such as: 1) the user's goal, s_u ϵ S_u; 2) the user's action au ϵ Au; 3) history / state of the dialogue s_d ϵ S_d. Thus, the POMDP state s is given by the tuple (s_u, a_u, S_d) and from the system's perspective; all those components are unobservable [32].

1) The user's goal, s_u gives the current goal or intention of the user. For example, user goal include a complete travel itinerary, a product the user would like to purchase or requesting information about a calendar.

2) The user's action au, gives the user's most recent actual action. For example, specifying a place the user would like to travel, responding to yes/no question, or a null response indicating the user took no action.

3) The dialogue history/state S_d, indicates any relevant history or state information. For example, particular slot has not been stated, if there any ungrounded items, a dialogue designer might wish to penalize asking an open question.

The POMDP action am ϵ Am is the action the machine takes in the dialog such as greeting the user or asking a question. At each time step t, the POMDP receives a single observation but it maintains a distribution over all possible user actions au. The factored POMDP is given by decomposing the POMDP transition function which is as follows:

$$p(s' \mid s, a_m) = p(s'_u, s'_d, a'_u \mid s_u, s_d, a_u, a_m)$$

$$= p(s'_u \mid s_u, s_d, a_u, a_m) p(a'_u \mid s'_u, s_u, s_d, a_u, a_m) p(s'_d \mid a'_u, s'_u, s_u, s_d, a_u, a_m)$$

The first term indicates the user goal model. At each time step t, it is assumed that the user's goal depends on the previous goal and the machine action.

$$p(s'_u \mid s_u, s_d, a_u, a_m) = p(s'_u \mid s_u, a_m)$$

The second term is the user action model which indicates what action the user is likely to take at each time step t. It is assumed that the user's action depends on the current goal and preceding machine action.

$$p(a'_u \mid s'_u, s_u, s_d, a_u, a_m) = p(a'_u \mid s'_u, a_m)$$

The third term is the dialogue model which indicates how the user and system actions affect the dialogue history. The current state or history of the dialogue depends on the previous history / state of the dialogue, user's action and system action.

$$p(s'_d \mid a'_u, s'_u, s_u, s_d, a_u, a_m) = p(s'_d \mid a'_u, s_d, a_m)$$

Thus, the transition function of POMDP is given by,

$$p(s' \mid s, a_m) = p(s'_u \mid s_u, a_m) p(a'_u \mid s'_u, a_m) p(s'_d \mid a'_u, s_d, a_m)$$

The observation function of POMDP is given by,

$$p(o' \mid s', a_m) = p(o' \mid s'_u, s'_d, a'_u, a_m)$$

The confidence score and rewards are not specified as this model is associated with a particular user goal and design objectives of the target system respectively. At each time t, the actions are selected depends on the belief state to maximize the cumulative long term reward by substituting and simplifying the above equations.

$$b'(s'_u, s'_d, a'_u) =$$
$$k \cdot p(o' \mid a'_u) p(a'_u \mid s'_u, a_m) \sum_{s_u \in S_u} p(s'_u \mid s_u, a_m) \sum_{s_d \in S_d} p(s'_d \mid a'_u, s_d, a_m) \sum_{a_u \in A_u} b(s_u, s_d, a_u)$$

Equation 4 Belief Update Equation

This model is tested with a simulated dialogue management problem in a travel domain in which the user is trying to buy a ticket to travel and compared the results with handcrafted policies and MDP baseline [33]. The results proved that POMDP maintains a well formed distribution over user goals and in case of certainty; it reflects in particular user goals. Since this model assumes the flat listing of flat components, the spoken dialogue systems with hierarchical components may result in poor performance.

3.2   Naturalistic Decision Making

The study of naturalistic decision making (NDM) has also evolved as a focused effort to describe how people make decisions in the real world. While some earlier researchers described decision making as being based on recognizing patterns in the situation that were matched to known patterns in memory, the area of NDM largely blossomed around the work of Gary Klein. NDM rejects certain previous research on decision theory (e.g., utility theory) as being largely normative instead of descriptive; therefore, such research fails to capture critical aspects of how people – particularly experts – actually make

decisions [35]. NDM specifically seeks to provide rich descriptions of how people make decisions in the real world, as opposed to within artificially contrived and constrained laboratory tasks. The environment in which NDM focuses may encompass ill-structured problems, uncertainty, time stress, risk, multiple and changing goals, multiple individuals and experienced decision makers.

3.2.1 Recognition-Primed decision (RPD)

Recognition-Primed Decisions (RPD) involves non-optimizing and non-compensatory strategies and requires little conscious deliberation. RPDs are marked by an absence of comparison among options. They are induced by a starting point that involves recognition matches that in turn evoke generation of the most likely action [35].

It has been tested applications of the model in a variety of tasks and domains, including fire ground command, battle planning, critical care nursing, corporate information management, and chess tournament play. These studies have shown good support for the validity and utility of the model presented in Figure 3.2 as it applies to individual decision makers. [35] Coding was evaluated as having 87% to 94% inter-rate reliability.
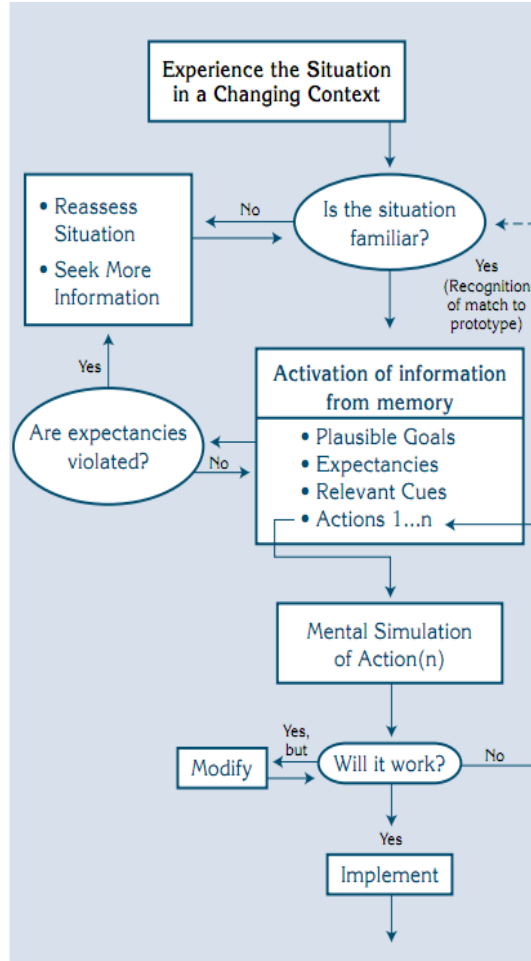
Figure 3.2: Complex RDP Strategy

3.2.2 Predicting User Mental States in SDS

[34] Propose a model for predicting the user mental state which can be integrated in the architecture of a spoken dialogue system as shown in Figure 3.3 As can be observed, the model is placed between the natural language understanding (NLU) and the dialogue management phases. The model is comprised of an emotion recognizer, an intention recognizer and a mental-state composer. The emotion recognizer detects the user emotional state by extracting an emotion category from the voice signal and the dialogue history. The intention recognizer takes the semantic representation of the user input and

28

predicts the next user action. Then, in the mental-state composition phase, a mental-state data structure is built from the emotion and intention recognized and passed on to the dialogue manager.
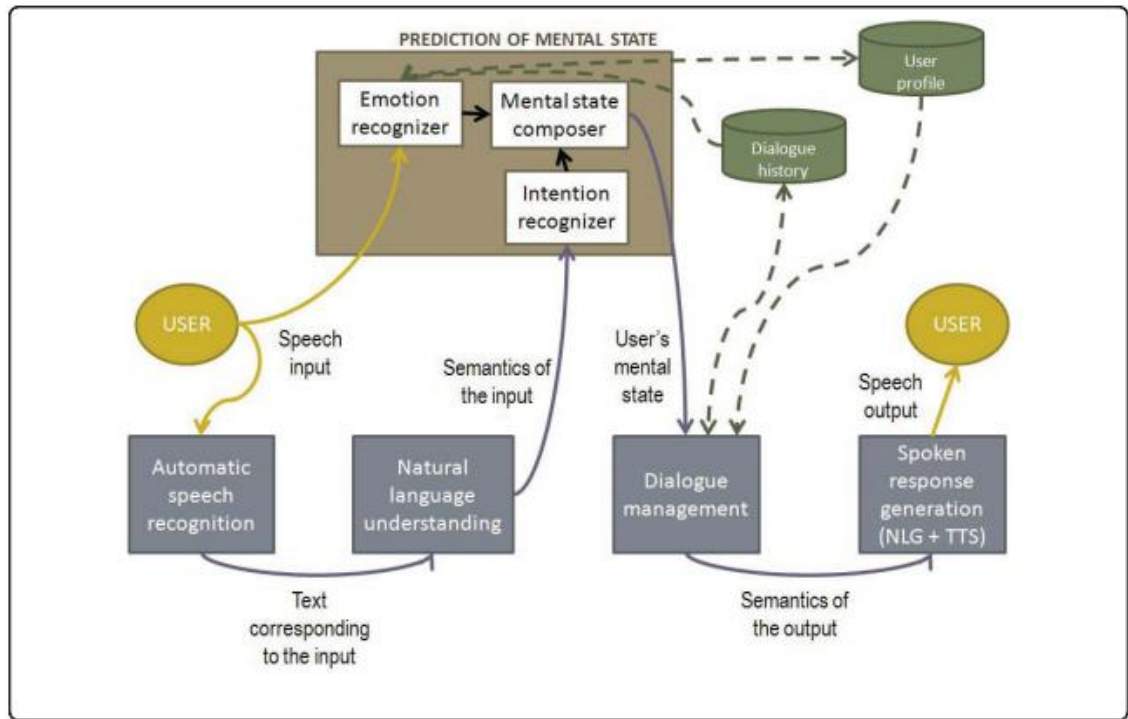
Figure 3.3 Integration of mental-state prediction into the architecture of a spoken dialogue system.

3.3 Contextual Control Model

Hollnagel in 1993 developed a Contextual Control Model (COCOM) to control and analyze team behavior based on cognitive modes. This model argued that the system decides what action to take next according to the context of situation. He observed that this approach is reactive both in the environment and individual perspective of the user.

The degree of control varies between four modes namely; scrambled opportunistic, tactical and strategic modes. He further argued that the team behavior should be analyzed as macro rather than micro level. These control modes of team behavior varies in terms of forward planning.

3.3.1 Testing COCOM by Assessing Team Behavior

In 2001, Stanton et al tested this COCOM with a team of people in a simulated energy distribution system. The results confirmed Hollnagel's model in two different ways. First, the team behavior could be categorized reliably into the four control modes provided a useful way of distinguishing between experimental conditions. Second, the progression between control modes conformed to the linear progression [36]. This model depicts the dynamism of the environment by determining how the operator should quickly shift to another mode depending upon the situation. If the action taken is correct then we can achieve the goal in short time and if the situation is already in a scrambled mode and the decision taken is incorrect, the goal will be removed and sets a panic situation in the environment. They explored the relationship between control modes and system states to see if different interfaces and proximity of personnel provide control teams with greater opportunity for strategic control and less demand for scrambled control. A framework is also set to transfer the control directly from scrambled to tactical and vice versa.
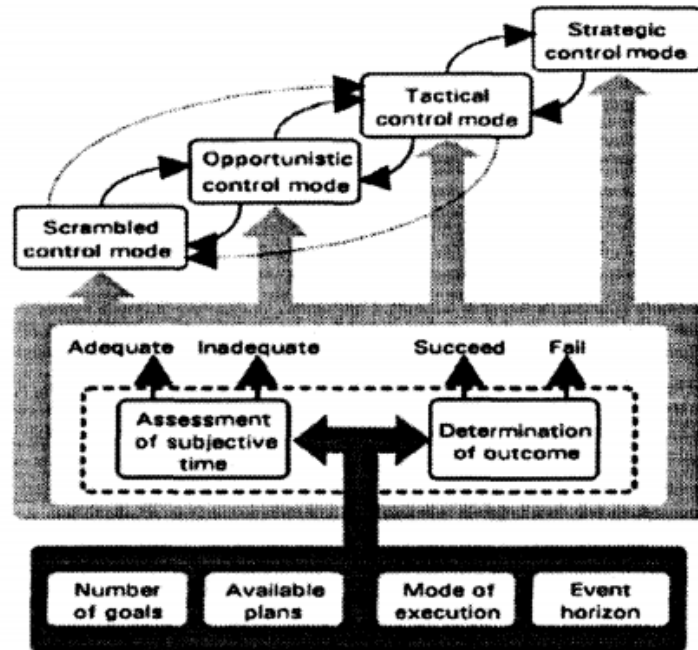
Figure 3.4 : Internal structure of Contextual Control Model [COCOM]

3.3.2 COCOM for Dynamic Decision Making

In 2006, Karen Feigh and Amy Pritchett introduced this COCOM in the design of support systems for dynamic decision making in Airline operations. They tested this model with the human operator and concluded that the regulation for dynamic systems has implication for both internal and external dynamic systems, for example: flight schedule. In the dynamic system, the individual's transition between COCOM controls modes to maintain the control over the dynamic condition, which in turn depends on the current context of the situation. The main feature of this model is availability of time. If there is time available is too short, then the control will be in opportunistic mode. There are several behaviors which they determine using this model namely, perception, situation assessment, communication, coordination, analysis, alternative generation and comparison of alternatives and tracked how these behaviors changes under different

contexts. Traditionally, support systems are designed to use single human activity, decision making and ignores several behaviors required to obtain successful goals. This analysis proved that, along with decision making other activities like judgment, coordination, information gathering, and solution generation can also be considered to achieve optimal solution for a particular situation [37]. They further extended their framework of COCOM to design and test multi-mode support systems for airline operations to improve airline recovery from irregular operations and airline rescheduling tasks [38]. It provides a useful framework to view the changes in cognitive work in response to contextual features such as time limit and information availability. Control in this model is conceptualized as planning what to do in the short-term and within the time horizon of the system with which the human is interacting.

3.4 Information based theory for Belief state history in POMDP based DM

Information space informally speaking is the space contained all the observations have been obtained, all the actions have been taken by the agent and the initial state. This space linearly grows with the new observation obtained and the actions applied. The way of manipulating this space has been divided mainly to three methods: traditional approach, nondeterministic approach and probabilistic approach. Under each approach, lots of strong assumptions have been made to make the method sufficiently generate policies. In the section 3.4.1 will give the overview of the history information space and then main dialogue management approaches except for POMDP-based approach will be analyzed upon history information space.

3.4.1 An Analysis with History Information Space

The formal definition of history information space is as following [20]: The set of all observation histories is denoted as Y_F, and is obtained by a Cartesian product of k copies of the observation space:

$$Y_k = Y \times Y \times Y \times Y......... \times Y$$

The set of all action histories is the Cartesian product of k-1 copies of the action space U. Planning under information space is based on the information state which is always known. I_q denotes the initial condition space, the above mentioned known state which means the initial state i_0, is given, then $I_q \; \varepsilon \; X$. At the stage k or time step k, the history space at stage k is expressed by the following:

$$I_k = l_0 \times U_{k-1} \times Y_k$$

With the definition of the observation history and action histories, the definition of the history information space is the union of each information stage I_k over all K ε {0} U N as the following:

$$I_{hist} = I_o \cup I_1 \cup I_2 \cup I_3 \cup ....I_k$$

Traditional approaches try to use the history information space to estimate the state and conduct action based on the estimated state. Now, the planning can be taken in the information space without knowing the exact state. The plan can be expressed as

π_t: I→U

3.4.2 POMDP based DM with Belief state history

Analyzing the belief history information shows that the compact of I_{hist} of history information space into a derived information space in a compressed form of N, X, or results in loss of important information. The consequence is inflexibility for human-robot interaction as in the FSM-based approach, incapable of handling any ambiguity as in the frame/Bayes/MDP-based approaches, and insufficiency in dealing with uncertainties as in the POMDP-based approach. To overcome the shortcomings while retaining the advantages of POMDP-based approaches, this paper proposes a modified planning strategy as illustrated below.

$$\pi_{new}: \quad I'_{k-1} \cup I_k \rightarrow U$$

In this approach, both I_k and I'_{k-1} are still in the form of belief state, and state updating still uses the existing POMDP models. The addition of in the modified approach, however, introduces an important element to dialogue management, i.e., the history of belief state or the dynamics of belief state. Although the historical information of observations and actions is not maintained explicitly in $I'k-1$, the union I_K and $l'k-1$ in above equation diminishes the negative effect of Markov assumption and allows POMDP-based dialogue management to plan for actions with not only the current belief state but also the updated history before reaching the current state.

CHAPTER 4

Previous work on History Information Space and Trend

Dialogue management is fundamentally a problem of planning under the influence of uncertainty. This chapter analyzes the usage of trend information in different fields and how those trending information is helping to reduce the uncertainty in the corresponding field. Trading strategies are discussed in this chapter. Different techniques used for detecting the changes in the trend and at which point are discussed. Finally an analysis of the belief state information in terms of trend used to find the knowledge of the user on domain is analyzed.

4.1 Trend analysis and trading strategy

This approach consists of three steps, namely partitioning, analysis and prediction. A modification of the commonly used k-means clustering algorithm is used to partition stock price time series data. After data partition, linear regression is used to analyze the trend within each cluster. The results of the linear regression are then used for trend prediction for windowed time series data. The approach is efficient and effective at predicting forward trends of stock prices. Using our trend prediction methodology, we propose a trading strategy TTP (Trading based on Trend Prediction) [40].

4.1.1 Methodology for Trend Analysis

Data mining approach in this methodology consists of the following steps
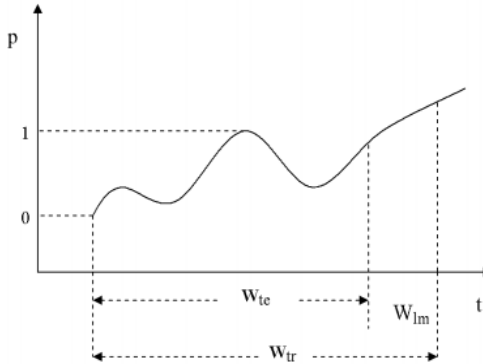
Figure 4.1 Schematic view of windowed time series

1. Initialization.

Select window lengths ω_{tr} and ω_{te} for training and test data respectively.

Select a test period.

Select training period.

2. Data Mining.

Create N training series of window length ω_{tr} from training period.

Normalize each series individually such that the first ω_{te} values of the series fall between 0 and 1.

Partition the training data into k clusters, which are represented by their cluster centers. We use the k-means clustering to group the training data based on attributes into k groups. k > 1 is a pre-specified integer number.

Classify all the clusters into two distinct classes using a linear regression model [6].A model is built based on the last ω_{tr} values of each cluster center. Class "UP" is labeled if the gradient is positive and "DOWN" otherwise.

3. Test models on test data.

Form a test series dataset with the window length ω_{tr} . Normalize them individually.

Consequently, values will fall between 0 and 1.

Assign a cluster label $c_i = j$ to time series i in test data such that cluster j (j =1, 2, $\cdots$, k)

has the smallest Euclidean distance to the normalized series i.

Assign the class ("UP" or "DOWN") of cluster j to time series i, where time series

i has cluster label j.

Calculate returns for a selected trading strategy.

4.2 1-2-3 Trend Change Method

[41] This method of finding the change in the trend is practiced in the stock market. This

method can be used where there are not a lot of data are available for mining but still

needs to draw trend line and find the strategy to analyze the movement of graph.

The procedure to this method is given below:

Draw a trend line from the highest high (Point C in the figure 4.2) to the lowest low (A)

on the chart such that price does not cross the trend line until after the lowest low (point

1), then follow these steps.
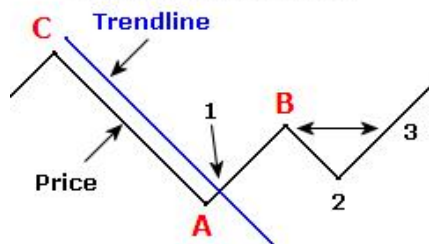
Figure 4.2 1-2-3 Trend change method

Step 1: Find where price closes above a down-sloping trend line. This is shown in the

chart as point 1 and a trend line pierce is the first indication of a trend change.

Step 2: Price tests a recent low. The recent low is at point A and the test is at point 2. Point 2 can be below point A but it must be clear that price is moving up, not continuing down.

Step 3: Price closes above a recent high. I show the high as point B and price completes the 1-2-3 trend change method when it rises above B at point 3. The high (point B) should be between points A and 2.

4.3 Frequent Pattern Mining

Frequent item sets play an essential role in many data mining tasks that try to find interesting patterns from databases, such as association rules, correlations, sequences, episodes, classifiers, clusters and many more of which the mining of association rules is one of the most popular problems. The original motivation for searching association rules came from the need to analyze so called supermarket transaction data, that is, to examine customer behavior in terms of the purchased products. Association rules describe how often items are purchased together. For example, an association rules "beer $\Rightarrow$ chips (80%)" states that four out of five customers that bought beer also bought chips. Such rules can be useful for decisions concerning product pricing, promotions, store layout and many others

Let I be a set of items. A *set $X = \{i_1, \ldots, i_k\} \subseteq I$* is called an itemset, or a *k*-itemset if it contains *k* items.

A transaction over I is a couple $T = (tid, I)$ where *tid* is the transaction identifier and I is an itemset. A transaction $T = (tid, I)$ is said to support an itemset $X \subseteq I$, if $X \subseteq I$.

A transaction database D over I is a set of transactions over I. We omit I whenever it is clear from the context.

The cover of an itemset *X* in *D* consists of the set of transaction identifiers of transactions

in D that support *X*:

$$cover\ (X,\ D) := \{tid\ |\ (tid,\ I) \in D,\ X \subseteq I\}.$$

The support of an itemset *X* in *D* is the number of transactions in the cover of X in D:

$$Support\ (X,\ D) := |cover(X,\ D)|.$$

The frequency of an itemset *X* in *D* is the probability of *X* occurring in a transaction

$$T \in D:\ frequency(X,\ D) := P(X) = support(X,\ D)\ /\ |D|.$$

Note that *|D| = support ({}, D).* We omit *D* whenever it is clear from the context.

An itemset is called frequent if its support is no less than a given absolute minimal

support threshold σ_{abs}, with $0 \leq \sigma_{abs} \leq |D|$.

4.3.1 The Apriori Algorithm

The first algorithm to generate all frequent itemsets and confident association rules was

the AIS algorithm by Agrawal et al. [44], which was given together with the introduction

of this mining problem. Shortly after that, the algorithm was improved and renamed

Apriori [44] by exploiting the monotonicity property of the support of itemsets and the

confidence of association rules. The same technique was independently proposed by

Mannila et al. [45]. Both works were cumulated afterwards.

4.3.2.1.1 Itemset Mining

Assume for simplicity that items in transactions and itemsets are kept sorted in their

alphabetical order unless stated otherwise. The itemset mining phase of the Apriori

algorithm is given in Algorithm 1 below. We use the notation *X[i]*, to represent the ith

item in *X*. The k-prefix of an itemset *X* is the k-itemset *{X[1], . . . , X[k]}.*

Algorithm 1 Apriori - Itemset mining

Input: $\mathcal{D}, \sigma$
Output: $\mathcal{F}(\mathcal{D}, \sigma)$
1: $C_1 := \{\{i\} \mid i \in \mathcal{I}\}$
2: $k := 1$
3: **while** $C_k \neq \{\}$ **do**
4:     // Compute the supports of all candidate itemsets
5:     **for all** transactions $(tid, I) \in \mathcal{D}$ **do**
6:         **for all** candidate itemsets $X \in C_k$ **do**
7:             **if** $X \subseteq I$ **then**
8:                 $X.support++$
9:             **end if**
10:         **end for**
11:     **end for**
12:     // Extract all frequent itemsets
13:     $\mathcal{F}_k := \{X \mid X.support \geq \sigma\}$
14:     // Generate new candidate itemsets
15:     **for all** $X, Y \in \mathcal{F}_k, X[i] = Y[i]$ for $1 \leq i \leq k-1$, and $X[k] < Y[k]$ **do**
16:         $I = X \cup \{Y[k]\}$
17:         **if** $\forall J \subset I, |J| = k : J \in \mathcal{F}_k$ **then**
18:             $C_{k+1} := C_{k+1} \cup I$
19:         **end if**
20:     **end for**
21:     $k++$
22: **end while**

Figure 4.3 Apriori - Itemset mining

The algorithm performs a breadth-first search through the search space of all itemsets by iteratively generating candidate itemsets C_{k+1} of size $k+1$, 13starting with $k = 0$ (line 1). An itemset is a candidate if all of its subsets are known to be frequent. More specifically, $C1$ consists of all items in $\mathcal{I}$, and at a certain level k, all itemsets of size $k + 1$ in $B_d{}^-(F_k)$ are generated. This is done in two steps. First, in the join step, F_k is joined with itself. The union $X \cup Y$ of itemsets X, Y $\in$ F$_k$ is generated if they have the same $k - 1$-prefix

(lines 20–21). In the prune step, $X \cup Y$ is only inserted into C_{k+1} if all of its k-subsets occur in F_k (lines 22–24). To count the supports of all candidate k-itemsets, the database, which retains on secondary storage in the horizontal database layout, is scanned one transaction at a time, and the supports of all candidate itemsets that are included in that transaction are incremented (lines 6–12). All itemsets that turn out to be frequent are inserted into F_k (lines 14–18). Note that in this algorithm, the set of all itemsets that were ever generated as candidate itemsets, but turned out to be infrequent, is exactly $B_d - (F)$. If the number of candidate $k + 1$-itemsets is too large to retain into main memory, the candidate generation procedure stops and the supports of all generated candidates is computed as if nothing happened. But then, in the next iteration, instead of generating candidate itemsets of size $k + 2$, the remainder of all candidate $k+1$-itemsets is generated and counted repeatedly until all frequent itemsets of size $k + 1$ are generated.[44]

4.4 Dialogue Management Approaches Analyzed Upon History Information Space

According to the theory of information space [42], the only information available to a decision process at stage K of a dialogue is the history of all observations Yk at that stage and the history of all actions U_{k-1} that have been taken before that stage. Let Y, U denote the observation space and the action space respectively. Given an initial condition η_0, Y_k and U_{k-1} are two Cartesian products of observation and action spaces respectively at their corresponding stages.

$$\tilde{Y}_k = Y \times Y \times Y \cdots \times Y$$

$$\tilde{U}_{k-1} = U \times U \cdots \times U$$

If ηo belongs to an initial condition space I_0, a history information space is formed as the union of I_0 and $I_K = Io \times \acute{U}_{k-1} \times \acute{Y}_K$ for up to the kth stage.

$$I_{hist} = I_0 \cup I_1 \cup I_2 \cup \cdots \cup I_k$$

An information-feedback plan $\pi = (\pi_1, \pi_2,\ldots)$ then maps I_{hist} into a sequence of actions

$$\mu_1, \mu_2, \mu_3, \ldots \in U$$

$$\pi: I_{hist} \longrightarrow U$$

An optimal plan $\pi *$ maximizes a given stage-additive cost function.

The history information space includes all the information which is so complicated. In the perspective of finding practical solution, it is not easy to manipulate the history information space under this complicated information space. In this case, the history information space is usually mapped to another derived space by the information mapping function to resolve the manipulation problem of the history information space. With the derived information space, some information loses result in the inappropriateness of the generated policies. In this section, different approaches will be discussed with the corresponding information mapping method upon the history information theory.

4.5 Analysis of history information space in dialogue management approaches with trend information

This section analyzes the belief state information generated by the POMDP model on each turn and extracts trend from the data. Information space informally speaking is the space contained all the observations have been obtained, all the actions have been taken by the agent and the initial state. This space linearly grows with the new observation obtained and the actions applied. The way of manipulating this space has been divided mainly to three methods: traditional approach, nondeterministic approach and probabilistic approach. Under each approach, lots of strong assumptions have been made to make the method sufficiently generate policies. The main dialogue management approaches except for POMDP-based approach will be analyzed upon dialogue history information space in terms of trend.

In section 4.3 dialogue management approaches is analyzed with history information space. The history information has the condensed form of the observation and action histories at the point K. This space has no information about the user's activity and about their belief from the point 0 to K.

In Section 4.1, 4.2 we have discussed about the trend analysis which seeks out and examines systematic historical patterns in financial statements or other quantitative data. Such analysis of data over time can vary from primarily descriptive techniques to more complex cause-and-effect methods. Trend analysis usually involves choosing one fiscal period as a base period and then expressing subsequent quantities as a percentage of the data associated with this base period. In the case of stock, changes in all items could be assessed in relation to the base period. Significant changes can then be investigated

further. Note that trend analysis can be performed to determine changes in the number of physical units.

Trend analysis is valuable when one wants to use historical data to predict future values or to calculate expected values for comparison to actual current values. Trend analysis is also useful for identifying unexpected variances that may indicate strategic or operational changes or entity weaknesses worthy of additional exploration and analysis.

4.6 Conclusion

In this chapter, we discussed briefly about the methods used to analyze the change in trend with small amount of data and also the dialogue management approaches analysis with history information space. This chapter also discusses about the belief history information loss in the information space which is useful for making decision based on the user's activity. From the literature, it is evident that the issues and limitations of both the research areas are still remaining unsolved. As the coin has two sides, each approach has its own drawback to be taken care of in future.

CHAPTER 5

The Proposed Method

This chapter gives a detailed explanation about the contribution of this thesis. Dialogue management is fundamentally a problem of planning under the influence of uncertainty. This chapter first uses the theory of information space to examine the POMDP-based approach of dialogue management, and then proposes a new approach for better recognition of user intention using the trend of belief state history and predicting next user belief state. The advantages of the new approach will be demonstrated with experiments in the next chapter.

5.1 Shortcoming with the Current Models

The POMDP-based approach avoids the need to estimate system state by using a set of probability distribution over belief state in the planning process. Together with the action at the Kth stage, previous belief state, the system uses new observations to update the belief state and plans for action at the next stage. In the process, the state of the system and the user is hidden in the information space. As defined in Eq. 3 for the flat model and Eq. 4 for the factored model, history information state is mapped to a probability distribution over the unknown system state. As this conversion is based upon on the Bayes filter theory, which in turn is under the Markov assumption, the POMDP-based approach plans for actions with only the current belief state, which is clearly illustrated in the b elements in both Eq. 3 and Eq. 4.

In POMDP models, actions A_m at the previous stage lead to observations probability Z at the *Kth* stage, which corresponds to the Cartesian product of U_{k-1} x Y_k in I_k. It is a simplification of I_{hist} into I_k, resulting a complete loss of history information, including changes in belief states, series of observations, and sequences of actions. The simplified version of Eq. 4 uses the following formula.

$$\pi' : \quad I_k \rightarrow U$$

Equation 5 Simplified version of eq: 4

Planning with POMDP models is better than all the other existing approaches as it does not rely on estimated system state, and is able to handle input uncertainty. However, the elimination of I_0 U I_1 U I_2 U I_3 U.... I_{k-1} from I_{hist} makes it impossible to trace changes in belief state and to retrieve the historical information of observations and actions. In other words, belief state is a static probability distribution over the current system state only. As a consequence, the POMDP-based approach is unable to deal with uncertainty in belief state itself, which corresponds to uncertainty in either user's actions or the observation of user's actions.

In a another perspective, the POMDP-based dialogue management approach only models the user's goal or it can be considered as a user modeling rather than a task modeling or machine state modeling. Although, when dealing with the observation uncertainties and action uncertainties, the POMDP-based approach outperforms than other approaches. This advantage is even more obvious when the error rate of the input is high. The POMDP based approach tries to listen correctly at its best. While, what if the user's goal

it's trying to listen is not correct itself at the beginning? The task will finally end up with the failure although DM listens correctly. Usually the dialogue systems make a assumption that the user can always answer the questions from the agent. However, in the real life condition, the user is always lack of domain knowledge and provides unreasonable information to the agent. This situation will be worst when the user can not actually understand the question generated by the agent. If the dialogue management approaches only models the user without its own domain knowledge level inference, the task cannot achieved. In the process of the human computer interaction, if the computer can appropriately influence the user and guide the user, the task is more probably to be achieved.

5.2 Change-Point Analysis for Detecting Changes in trend

Change-point analysis is a powerful new tool for determining whether a change has taken place. It is capable of detecting subtle changes missed by control charts. Further, it better characterizes the changes detected by providing confidence levels and confidence intervals. When analyzing historical data, especially when dealing with large data sets, change-point analysis is preferable to control charting. A change-point analysis is more powerful, better characterizes the changes, controls the overall error rate, is robust to outliers, is more flexible and is simpler to use.

There are numerous approaches to performing a change-point analysis. The one used in this paper has been implemented in Taylor [43]. The procedure used by Taylor [43] for performing a change-point analysis iteratively uses a combination of cumulative sum charts (CUSUM) and bootstrapping to detect the changes

Bootstrapping results in a distribution free approach with only a single assumption, that of an independent error structure. Both control charting and change-point analyses are based on the mean-shift model. Let X_1, X_2, ... represent the data in time order. The mean-shift model can be written as

$X_i = \mu_i + \varepsilon_i$

Where μ_i is the average at time i. Generally $\mu_i = \mu_{i-1}$ except for a small number of values of i called the change-points. ε_i is the random error associated with the i-th value. It is assumed that the ε_i are independent with means of zero. [43] Provides a procedure for detecting a departure from this assumption. Once a change has been detected, an estimate of when the change occurred can be made. One such estimator is the CUSUM estimator. Let m be such that:

$$|S_m| \;=\; \max_{i=0,\dots,N} |S_i|$$

5.3 Belief State trend analysis - The proposed work

This section discusses the existing POMDP models with the limitations and the summary of the methods and terminologies involved in my proposed frame work. This section also contains the mathematical derivations and algorithmic approach to discuss my proposed work.

5.3.1 Method description

As discussed in the chapter 3, POMDP models are problem of planning under uncertainty. In order to improve the efficiency of the POMDP models, the uncertainty

about the environment has to be reduced by providing the maximum information that can be prepared, In POMDP-based approach for dialogue management machine state is calculated with the observation and transition probability distribution defined in the domain as POMDP specification. The dialogue management agent should be given adequate information about the environment to make better decisions against the world. The information focused in this thesis is the user activity, knowledge and intention.

In section 4.4 dialogue management approaches analysis with history information space states that the Cartesian product of information at all the time or all the point till k are carried but in the condensed form. This let to loss of information related to the belief history. Belief history is the only evidence as how the user had conversed. Condensed form of information which is I_{hist} drops belief history in turn user's activity. Recording the user's activity will led to the decision making related to particular user not in a generic way as the dialogue management with just the POMDP model mentioned.

Though all the POMDP techniques have better approach by overcoming issues of the previous models, they have their own limitations. As evident from the literature, it is clearly know that these approaches fail to handle uncertainty and predicts the real-world state as static. And the decision made by the machine depends only on the current state alone in long- term and short-term goals. These models were developed to handle extremely large systems with millions of dialogue state and complex applications but none of the models concentrate to overcome the POMDPs natural property of predicting static belief states.

To provide innate interaction between the human and machine, it is always intelligent to equally concentrate more on both the technical and decision making mechanism. The

machine should be trained to handle random situations and able to hold overall control of the system in order to provide a more natural way of service to the users. We have proposed a framework to provide a dynamic system by modifying the POMDP model and integrating Contextual Control Modes for dynamic decision making mechanism by analyzing the user's activity and inferring their knowledge in the domain and the user's goal stability in terms of trend in the belief history on all the states

We have made **two contributions** in this thesis. First, we have analyzed the belief history in terms of trend to infer the user's activity and extract the rate of change in the trend which in turn indicates the significant change in user's intention and making the decision accordingly. And also customized the mathematical evaluation of POMDP model depending upon the framework proposed. Second, we have incorporated Contextual Control Modes in Dialogue Management to handle the decision making mechanism in the dialogue manager.

5.3.2 Modified POMDP Approach with belief trend

Analysis in the previous subsections shows that the compact of I_{hist} of history information space into a derived information space in a compressed form of I_k results in loss of important information. The belief states $B'(s)$ calculated for every iteration and the action A_m is chosen from the set of actions against the environment. To overcome the shortcomings while retaining the advantages of POMDP-based approaches, we propose a modified planning strategy as illustrated below. The modified approach concentrates on the belief state values and these values are recorded for analysis. This information is stated as belief history B_{hist} of a user.

$$B_{hist} = B_0 \ X \ B_1 \ X \ B_2 \ X .... \ B_k$$

Equation 6 Belief history

B_0 is the initial belief calculated by the POMDP model when the agent initiates the conversation with the user. $B_1, B_2 .. B_k$ are the belief calculated in the subsequent iterations. Referring to the equation derived in the section 4.2

$$I_K = I_o \ x \ Ú_{k-1} \ x \ Ý_k$$

The above equation represents the only action history, observation history and initial information available in the space. Now along with the belief trend information the equation is derived as follows

$$I_k = I_o \ x \ Ú_{k-1} \ x \ Ý_k \ x \ T`_k$$

Equation 7 Derived equation with belief trend information

Where $T`_k$ = Trend(B_{hist})

In the equation 7 $T`_k$ is the trend information available with B_{hist} up to k^{th} point.

An information -feedback plan $\pi = (\pi_1, \pi_2, \pi_3, \pi_4.....)$ then maps I_{hist} in to a sequence of actions $\mu_1, \mu_2, \mu_3, \mu..... \in Ú$

$$\pi^{T`} : I^{T`}_{hist} \rightarrow U$$

Equation 8 Optimal plan with belief trend

Where $I^T{}_{hist}$ is the history information space with belief trend up to the kth stage. π^T is the optimal plan with belief trend information $\pi^{T`}*$ maximizes a given stage-additive cost function.

In the new approach, $T`$ is referred as the belief trend information which is evaluated with the history of belief states B_{hist} up to the current stage. Belief states are recorded every iteration and analyzed for trend which is referred as belief trend are added to the previous information available in the information space. The addition of $T`$ in the modified approach, i.e trend information about the belief history or the dynamics of the user's activity. The trend of belief history is added along with the observation and action history information which are considered already in the in the information space.

The equation derived in Eq: 6 diminishes negative effect of Markov assumption and allows POMDP-based dialogue management to plan for actions with not only the current belief state but also the updated trend of belief history before reaching the current state. After every user action which is observation by the POMDP model agent, belief is calculated by the agent and these values are recorded and mining is done with the current belief and trend $T`$ is estimated by the agent with the belief history till that current stage. The trend information about the belief history reveals the information on user's activity with the conversing agent as belief are calculated by the agent at every stage depending upon the conversation which are the user's action. This implies that the analyzing the user's activity can be done with the agent's belief trend history at any stage. This reduces the uncertainty caused due to particular type of user.

The uncertainties that the original POMDP-based approaches fail to handle are mainly rise from situations in which the agent cannot differentiate the type of users and different

modes to treat them to achieve their goal. Every user has different level of knowledge about the domain and different strategy of approach to reach their goal. User with higher level of knowledge about the domain shouldn't be treated same as the user who has lesser level of knowledge. The rate of change in the user intention indicates the user's knowledge about the domain. The rate of change in the user's intention can be observed from the rate of change of trend in belief history.

Depending upon the belief trend, one of the four modes discussed in section 3.6 from COCOM is chosen to handle the particular type of user. This switching mode let the POMDP model to choose a complete different policy to treat the particular user. This helps the agent to make decisions dynamically with the user's activity choosing different policy at different stage. This reduces uncertainty about the user knowledge and helps the agent to treat high level knowledge user to achieve their goal in short time also lesser level knowledge user to achieve the goal independent of the time factor with detailed explanations about the domain. In precise different actions are made by the agent depending upon the world. This modified approach led the agent to choose the optimal policy depending upon the user. Thus, this approach concentrates not only the immediate reward but also on the significant increase in the future reward using different policies on different modes.

The figure 5.1 shown below is the architecture of the POMDP based dialogue management system with our modified approach, in which direct flow from state estimator to policy is broken and the belief state at every stage is recorded into the belief data store. Every time state estimator updates the belief state, it signals trend analyzer to generate the rate of change of trend in the belief history of the current user. The recent

changes in the belief trend determine the mode of operation which in turn selects the

optimal policy π^T that takes suitable action on the environment.
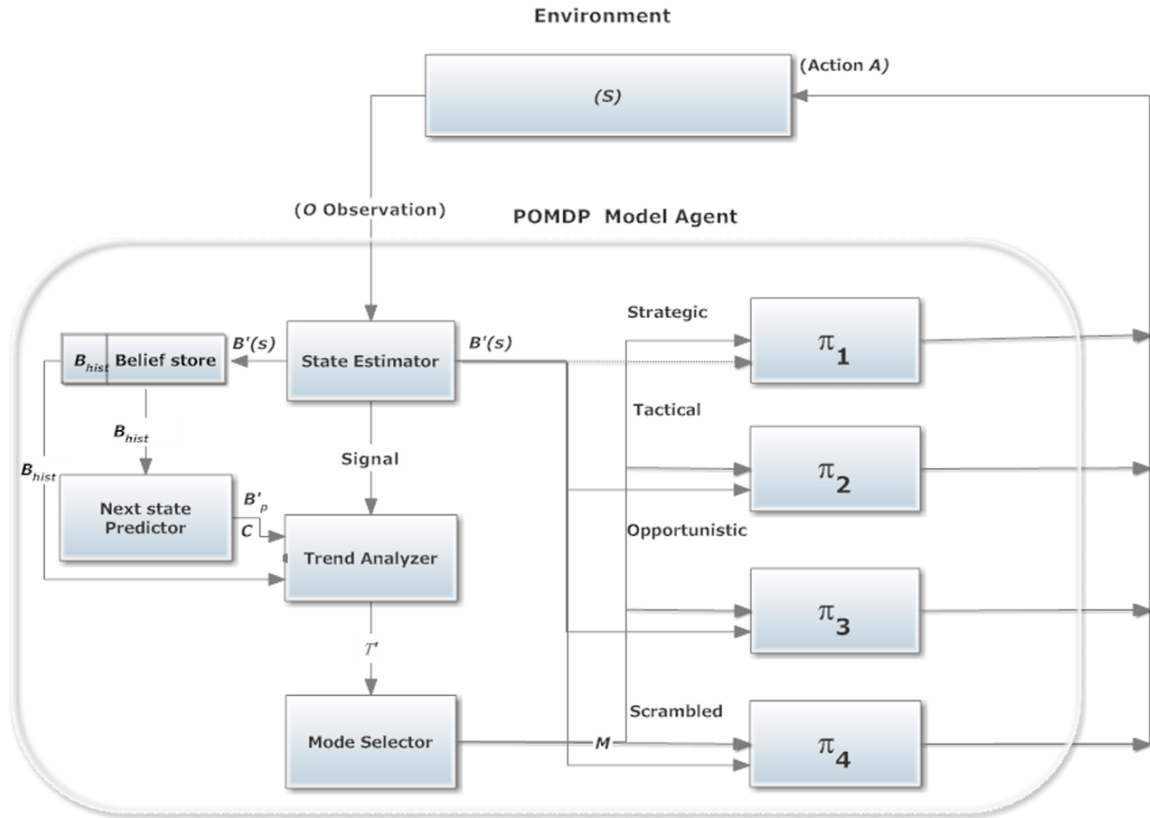
Figure 5.1 Architecture of the proposed approach

The next predictor module analyzes the belief history of all the users interacted with the

dialogue management system and predicts the next belief state using the frequent pattern

mining technique.

This technique is compared to the naturalistic decision making model. The belief pattern

exist in the belief history of a particular user are analyzed with the pattern in the memory.

The predicted next belief state with the confidence is send to the trend analyzer. It then

finds the rate of change of belief trend that chooses the mode M accordingly which in

turn selects the policy. The selected optimal policy π^{T} maps the belief and the action to execute on the environment.

The structure of the proposed system has 4 basic components they are: dialogue interface, I/O controller, dialogue manager and knowledge base. The proposed system is plugged into the dialogue manager. It has trend analyzer and predictor component which processes the belief values and selects the policy corresponding to the mode derived by the analyzer. Dialogues are generated by the I/O controller based on the policy chosen by the policy selector subcomponent.
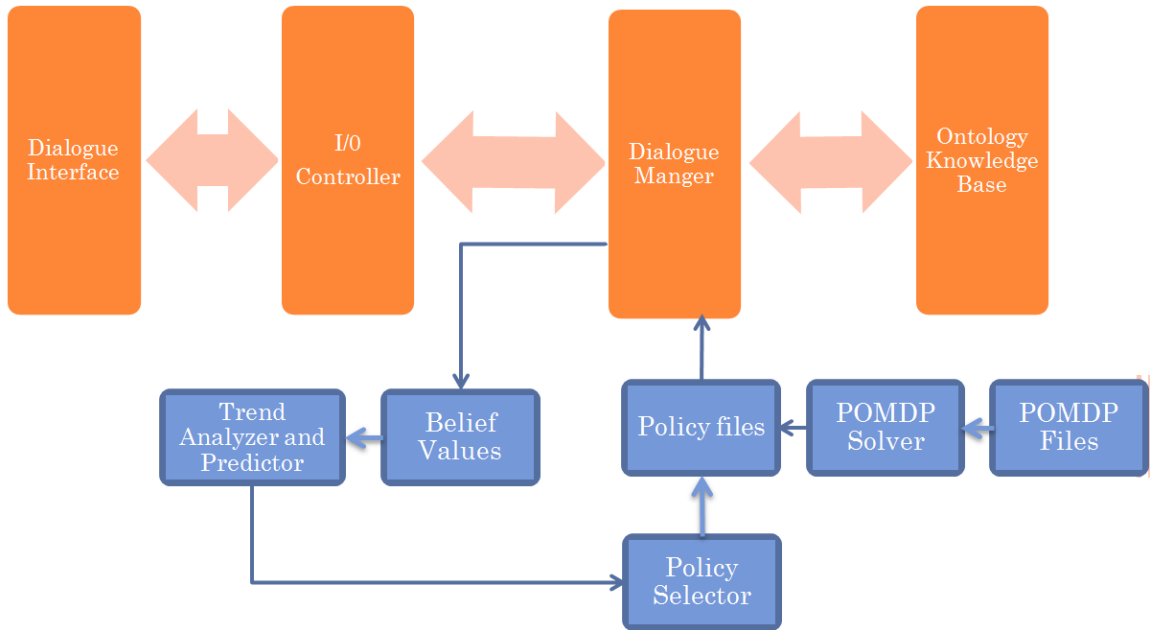
Figure 5.2 Structure of the proposed diagram
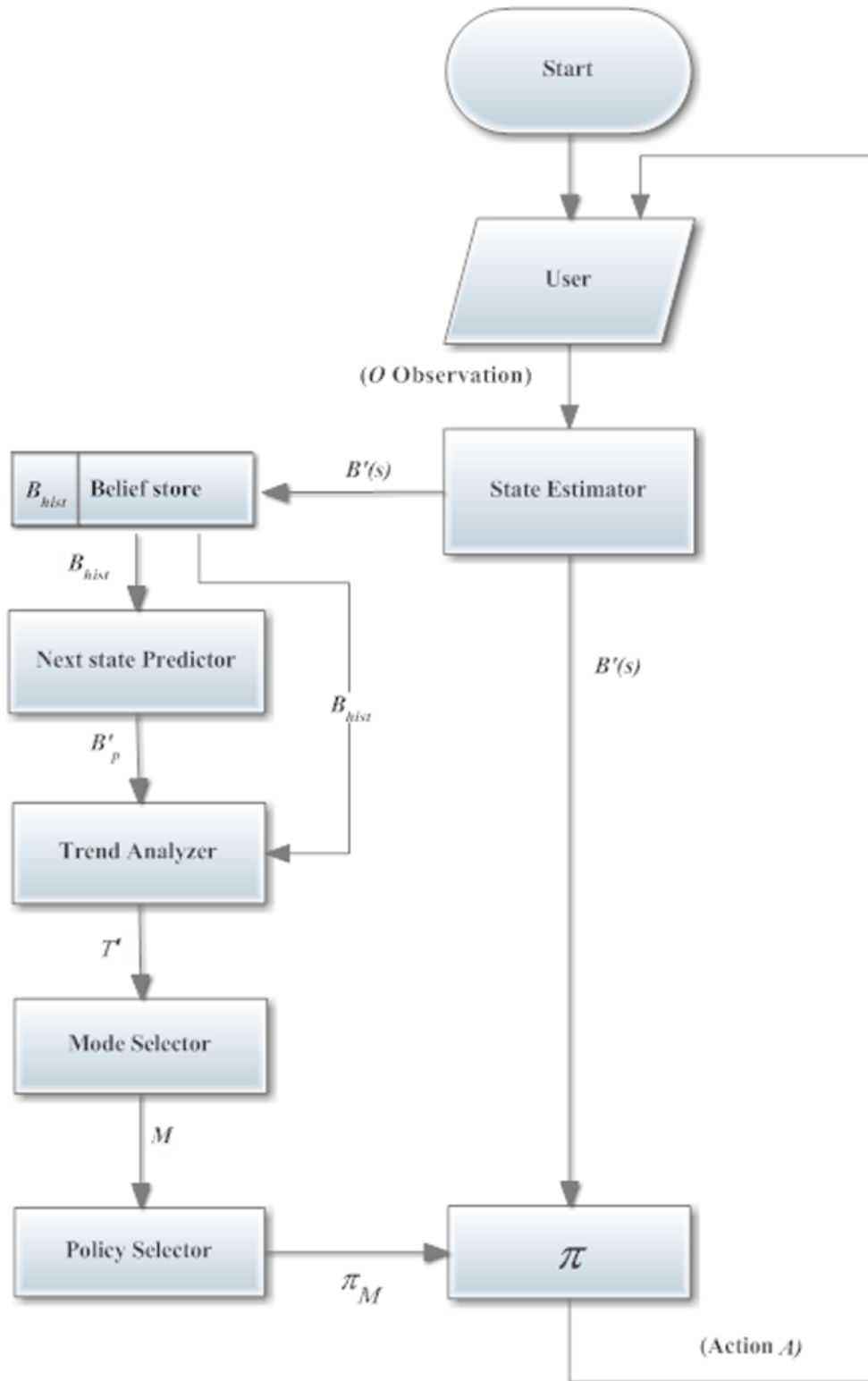
Figure 5.3 Flow chart of the proposed work

1. **New_proposed_Method (*Bel(s), o, a*)**

2. FOR ALL *s* DO

3.   *Bel'(s)* $= \alpha P(O_{t+1}|S_{t+1}, a_t) \sum P(S_{t+1}|S_t, a_t) b_t(S_t)$

4.   IF *Bel'(s)* < 0.05

5.   Delete *Bel'(s)*

6.   CONTINUE

7.   *Bel_history* = Append_to_belief_history(*Bel'(s)*)

8.   *Bel_precdicted, Prediction_Confidence* = Predicted_next_state (*Bel_history*)

9.   *Bel_trend_roc* = Get_ROC_belief_trend (*Bel_history* , *Bel_precdicted,*

    Prediction_Confidence)

10.   *COCOM_mode* = Select_mode (*Bel_trend_roc*)

11.   IF *COCOM_mode* =strategic

12.   *Machine_next_action* = Select_policy (*policy_startergic, Bel'(s)*)

13.   ELSE IF *COCOM_mode* =tactical

14.   *Machine_next_action* = Select_policy (*policy_tactical, Bel'(s)*)

15.   ELSE IF *COCOM_mode* =*opportunistic*

16.   *Machine_next_action* = Select_policy (*policy_opportunity, Bel'(s)*)

17.   ELSE IF *COCOM_mode* =*scrambled*

18.   *Machine_next_action* = Select_policy (*policy_scrambled, Bel'(s)*)

19.   *RETURN Machine_next_action*

20. **Predict_next_state (*Bel_history*)**

21. *Bel_predicted* = Frequent_Pattern_Mining(*Bel_history*)

22. *Prediction_Confidence* = Get_Predicted_Confidence (*Bel_predicted*)

23. RETURN *Bel_predicted, Prediction_Confidence*

Figure 5.4 Pseudo Code   of Modified Approach

In the pseudo code the *Bel(s), o and a* are the inputs to the proposed method. *Bel(s)* is the previous belief state of the last stage, *o* is the latest observation and *a* is the last action taken by the machine. By recording the previous belief state, the belief state will be updated based on the POMDP theory and for all the belief states with possibility less than 0.05. The other belief states are stored in the belief store calling function Append_to_belief_history (). The updated belief history is sent to the Predicted_next_state () function to get the predicted next belief state using frequent pattern mining along with the confidence. The results of the prediction are sent to the trend analyzer to get the rate of change in the belief trend calling the function Get_ROC_belief_trend (). Rate of change of the belief trend is used to select one of the COCOM modes. Select_policy () in turn select the optimal policy with *Bel'(s)*. Every policy is mapped to a machine action i.e. *Machine_next_action* which is returned by our proposed method to execute against the environment. In the algorithm it is clearly mentioned that the program runs for all the states which is the number of count of states specified in the POMDP file N. Thus the time complexity calculated to run the proposed algorithm is O (N).

CHAPTER 6

Experiments and Discussions

In this chapter, the domain background applied in our case study will be introduced first and the implementation platform, utilized tools and corresponding details will be explained after. The results for the Belief Trend Analysis and prediction are explained first to understand the efficient trend induction to the POMDP mode for intention change. Finally, the results under different possible scenarios, analysis of the outcome and comparison with the baselines will be given. At the end of this chapter, the results under three scenarios and results analysis will be given.

6.1 Applied Domain Background

6.1.1 Ontology-based requirement model

A frame-based dialogue system is developed by Xieshen Zhang in his thesis "PROPOSED ONTOLOGY-BASED REQUIREMENT MODEL" which takes the instantiated ontology model as knowledge base. It is applied to elicit users' demands through human-machine interaction. Though to maintain the completeness and consistency of the customized requirements is very complicated and requires ontology reasoning, interactions for requirement elicitation are actually a group of slot-filling tasks. Questions such as whether users need a specific requirement will be proposed by the machine, and users will respond with their decisions on the very requirement. Therefore, users know what they are going to do and how it is going to be done, which means the requirement elicitation process can be modeled as a set of slot-filling subtasks, while the utterances, slots as well as value options for each slot will be retrieved from the

knowledge base, hence a framed-based dialogue system is capable of handling the interactions for requirement elicitation, in spite of its limited communication ability.

Fig. 6.1 depicts the interface of the dialogue system proposed by implemented by [49] Xieshen Zhang. It is divided into three parts. The utterances generated by the dialogue manager are displayed in the upper left textbox. Users can type their response in the lower left textbox. Meanwhile, the three lists on the right side contain the selected, dropped and to-be-evaluated requirements respectively.
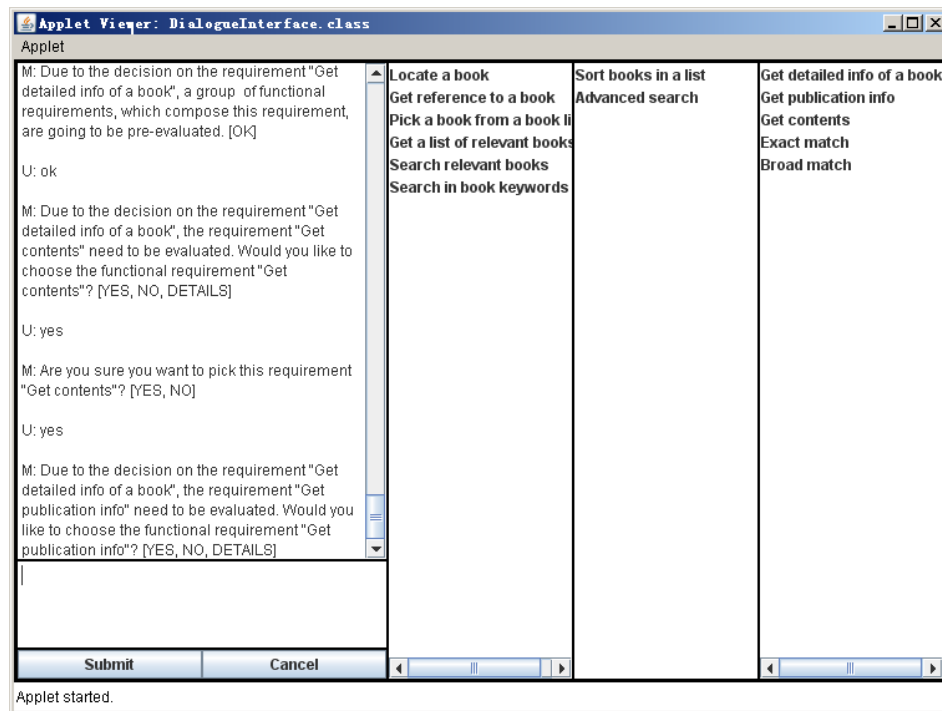
Figure. 6.1: Interface of the proposed dialogue system

An online book shopping system is used as cases study in Zhang's [49] this research. The structure of a typical but simplified online book shopping system is illustrated in Fig. 6.2. There are basically four modules: book locating, cart management, account management and order placing. Book locating module is responsible for book searching and retrieving book information; cart management module provides a list where users can save the

references of the books they want to buy; account management module manages users' personal, delivery and payment information; order placing module gathers information such as shopping list, payment, delivery, and total price, and helps users to place the order. It is assumed that account management is not necessary for an online book shopping system. Users can specify necessary information for each purchase without having it saved in the online bookstore.
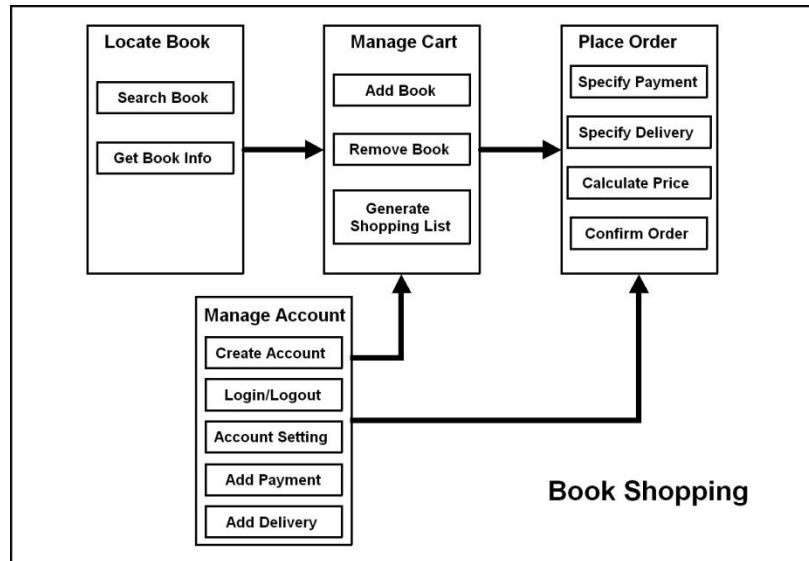
Figure. 6.2: Functionalities of an online book shopping system

6.1.2 Implemented Domain overview

Domain knowledge from Zhang's [49] method is partially used in our experiment based on a simulated situation in which an agent provides assistance to users in need of software to manage online book shopping. Software is an integration of different smaller modules developed to perform specific tasks. The smaller modules can be used independently with the base module or dependent to other modules which all together built to the base module. In the online book shopping domain, we have 6 software modules and every software module is dependent of each other. They are customized

with Locate a book, List of relevant books, Advanced Search, List of book references, locate a book, Get detailed info a book, Book publication info and get publication info. The user is allowed to select any module to build the software with the modules needed. The user is not expected to have the domain knowledge and may make unreasonable request. The user, however, is expected to be rational who is ready to change the goal after an explanation by the agent. For any software to construct the user has to select different modules which are integrated together and form complete software. It is assumed that the software constructed with any one of the 6 customizations as mentioned below.
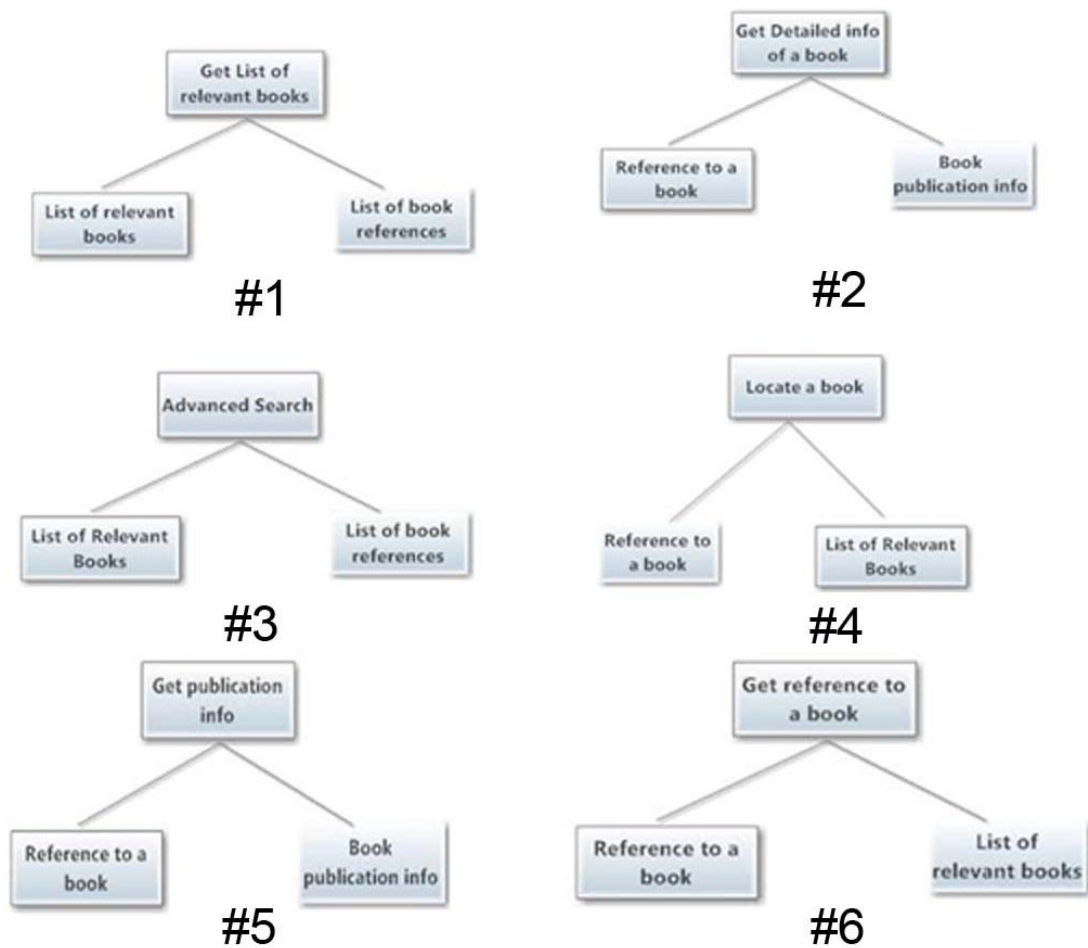
Figure 6.3 Software requirement customization

The user is allowed to choose the modules in any order to construct the software. Eg: As mentioned in Software requirement customization the user might choose Get publication info →Book publication info → Reference to a book or Book publication info →Reference to a book → Book publication info or the any other way. When the user goal is straight and the user has domain knowledge, the change in the intention of the user will be less which is determined by the change in the trend of belief state by the machine. This user will be considered to have domain knowledge and treated with strategic mode. Where in the explanation are short and technical words to explain. Pre- evaluations of modules are not done to save the overall user time to get the software customization task done. Scrambled mode is adapted when the change in the user intention is higher compared to that of the total transactions.

6.2   Implementation setup

Experiments are programmed in Eclipse IDE with Java 1.6. The setup also needs a Linux machine to generate alpha file from POMDP file using a POMDP solver. In this section the detail implementation and experimentation results of the Belief trend analysis and prediction is done to find the rate of change in the intention of user, thereby choosing the appropriate mode of operation. And also the techniques are discussed, analyzed and the results are shown. There are six main implementation required for this analysis:

- Agenda -based User behavior Simulator

- Belief History dataset generation

- POMDP Problem Specification

- POMDP Dialogue Management System - Ontology integrated

- ROC of User Intention with Belief Trend Analysis

6.2.1 Agenda -based User behavior Simulator

This section discuss about the simulator which does the simulation of user behavior based on agenda created. Agenda acts as a user goal at a particular point. Actions are taken based on the agenda.

6.2.1.1 User Simulation-Based Training

In recent years, a number of research groups have investigated the use of a two-stage simulation based setup. A statistical user model is first trained on a limited amount of dialogue data and the model is then used to simulate dialogues with the interactively learning DM [50]. The simulation-based approach assumes the presence of a small corpus of suitably annotated in-domain dialogues or out-of-domain dialogues with a matching dialogue format. In cases when no such data is available, handcrafted values can be assigned to the model parameters given that the model is sufficiently simple but the performance of dialogue policies learned this way has not been evaluated using real users.

6.2.1.2 User Simulation at a Semantic Level

Human-machine dialogue can be formalized on a semantic level as a sequence of state transitions and dialogue acts. At any time t, the user which is in a state S takes an action a_u, now the transaction of state in to the next state which is intermediate state S'. Now the user receives a system action a_m, and transactions into the next state S '' here the cycle go again right from the state S thus restarts.

$$S \rightarrow a_u \rightarrow S' \rightarrow a_m \rightarrow S'' \rightarrow \cdots$$

Assuming a Markovian state representation, user behaviour can be decomposed into three models: $P(a_u|S)$ for action selection, $P(S'|a_u,S)$ for the state transition into S', and $P(S''|a_m,S')$ for the transition into S''.

6.2.1.3 Goal- and Agenda-Based State Representation

Agenda-based methods to dialogue management (Wei and Rudnicky, 1999) this approach describe here factors the user state into an agenda A and a goal G

$$S = (A, G) \text{ and } G = (C, R)$$

During the course of the dialogue, the goal G ensures that the user behaves in a consistent, goal-directed manner G consists of constraints C which specify the require venue, eg. a centrally located bar serving beer, and requests R which specify the desired pieces of information eg. The name, address and phone number the user agenda A is a stack-like structure containing the pending user dialogue acts that are needed to elicit the information specified in the goal. At the start of the dialogue a new goal is randomly generated using the system database and the agenda is initially populated by converting all goal constraints into inform acts and all goal requests into request acts. A bye act is added at the bottom of the agenda to close the dialogue.

6.2.1.4 User Act Selection

At any time during the dialogue, the updated agenda of length N contains all dialogue acts the user intends to convey to the system. Since the agenda is ordered according to priority, with A[N] denoting the top and A[1] denoting the bottom item, selecting the next user act simplifies to popping n items off the top of the stack. Hence, letting au[i] denote the ith item in the user act au.

Agenda-based user simulator is used to select the user goal and communicate with the POMDP-based dialogue management system and performs the action. This simulator is made to run N number of times specified in the configuration. Belief states are generated which are recorded for every iteration and also for every user. These belief states are stored in the repository and they are finally used for the frequent pattern analysis.

6.2.2 Belief History dataset generation

This ontology integrated POMDP-based system is implemented in such a way that the belief of the machine's state is recorded at all the iterations. The belief recorded will have the machine's belief calculated with all the attributes at that stage. Below is the sample CSV file which has the belief history data that is used to extract the ROC of user's intention by calculating the ROC of belief trend.

Steps to prepare dataset for analysis:

- ➢ Run the agenda-based simulator programmed in Java using command line options as follows

```
pomdp_tool project.pomdp.dialoguesimulation.problemsim.PomdpModelCus
"\qa-dialogue-simple-swCustomization_strategic.POMDP"
"\qa-dialogue-simple-swCustomization_strategic.alpha"
 "\qa-dialogue-simple-swCustomization_tactical.POMDP"
"\qa-dialogue-simple-swCustomization_tactical.alpha"
"\qa-dialogue-simple-swCustomization_opportunistic.POMDP"
"\qa-dialogue-simple-swCustomization_opportunistic.alpha"
"\qa-dialogue-simple-swCustomization_scrambled.POMDP"
"\\qa-dialogue-simple-swCustomization_scrambled.alpha"  1000
```

➢ The simulator uses the POMDP and alpha file to calculate the belief state at every stage.

➢ The agenda-based simulator picks up the random goal and uses the corresponding agenda to achieve the user goal.

➢ The user goal can be changed randomly for any user or it might not be, this property makes the user different from one another.

➢ Belief state values are recorded and data set is stored in the local file system which is shown below.

```
1
------------------
StartState:fromLocateABooktoListOfRelavantBooks-fromLocateABook-
sn
Action:hello
Observation:fromLocateABook; Reward:0.0;
State:fromLocateABooktoListOfRelavantBooks-fromLocateABook-sn
mode:1
Action:askto
Observation:ListOfRelavantBooks; Reward:-1.0;
State:fromLocateABooktoListOfRelavantBooks-ListOfRelavantBooks-ss
mode:1
Action:fromLocateABooktoListOfRelavantBooks
Observation:null; Reward:10.0;
State:fromLocateABooktoListOfRelavantBooks-null-f
mode:1
Action:bye
Observation:toAdvancedSearch; Reward:1.0; State:end
Total reward:10.0
1
------------------
StartState:fromListOfRelavantBookstoLocateABook-toLocateABook-ns
Action:hello
Observation:toLocateABook; Reward:0.0;
State:fromListOfRelavantBookstoLocateABook-toLocateABook-ns
mode:1
Action:askfrom
Observation:ListOfRelavantBooks; Reward:-1.0;
State:fromListOfRelavantBookstoLocateABook-ListOfRelavantBooks-ss
mode:1
Action:fromListOfRelavantBookstoLocateABook
Observation:null; Reward:10.0;
State:fromListOfRelavantBookstoLocateABook-null-f
mode:1
Action:bye
Observation:fromLocateABooktoAdvancedSearch; Reward:1.0;
State:end
Total reward:10.0

Simulation output:
Total runs: 1000
Total turns: 402
Avg turns per run: 4
Avg reward per run: 10.0; Standard deviation: 0.8944271909999159
95% conf interval of avg reward/run: [ 9.445628283549746 ,
10.554371716450254 ]
Avg reward per turn: 6.5
```

Figure 6.4 Agenda-based simulator run of 1000 users

The figure 6.4, above shows the performance analysis report of the POMDP-based dialogue management system with belief trend. The simulation output shows that on run for 1000 users and the total dialogue turns found to be 402. The average turns for every user is 4. The average reward gained every run is 10.0. Standard deviation is calculated to 0.8944 with 95% confidence.

.

A1	A2	A3	A4	A5	A6	A7	A8
1	0	0	0	0	0	0	0
0.05	0.28	0	0.4	0.27	0	0	0
0.18	0.14	0.18	0.18	0.18	0.04	0.04	0.04
0.17	0.15	0.18	0.18	0.18	0.05	0.05	0.05
0.12	0.12	0.12	0.12	0.12	0.12	0.12	0.12
0.12	0.12	0.12	0.12	0.12	0.12	0.12	0.12
0.12	0.14	0.16	0.15	0.12	0.21	0.05	0.06
0.16	0.14	0.17	0.16	0.16	0.07	0.07	0.07
0.06	0.29	0	0.05	0.07	0.41	0.04	0.08
0.05	0.27	0	0.3	0.2	0.04	0.09	0.04
0.09	0	0.51	0.06	0.12	0	0.09	0.13
0.16	0.14	0.17	0.17	0.17	0.06	0.06	0.06
0.15	0.14	0.17	0.19	0.14	0.12	0.04	0.05
0.12	0.12	0.12	0.12	0.12	0.12	0.12	0.12
0.12	0.14	0.16	0.15	0.12	0.21	0.05	0.06
0.16	0.14	0.17	0.16	0.16	0.07	0.07	0.07
0.09	0	0.54	0.07	0.11	0	0.07	0.12
0.08	0.36	0	0.05	0.11	0.22	0.07	0.11
0.03	0.2	0	0.03	0.05	0.59	0.04	0.07
0.06	0	0.43	0.04	0.09	0	0.15	0.23
0.05	0.27	0	0.32	0.21	0.04	0.07	0.04
0.08	0.36	0	0.05	0.11	0.22	0.07	0.11
0.12	0.12	0.12	0.12	0.12	0.12	0.12	0.12
0.07	0	0.53	0.06	0.1	0	0.08	0.15
0.06	0.26	0	0.34	0.23	0.03	0.06	0.03
.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.

Table 6.1 Sample Belief recorded data

6.2.3 Dataset Analysis in Weka

Belief dataset generated are converted into two different classes TRUE and FALSE by analyzing the trend at every stage. The significant change in trend is marked as TRUE

and which doesn't have the significant change are marked as FALSE. This trend analysis is done using change point analysis for detecting trend change discussed in the section 5.2. This analysis creates another set which has two classes as show below.

A1	A2	A3	A4	A5	A6	A7	A8
FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
TRUE	FALSE	TRUE	TRUE	FALSE	FALSE	FALSE	FALSE
FALSE	FALSE	TRUE	FALSE	FALSE	FALSE	FALSE	FALSE
TRUE	FALSE	TRUE	FALSE	FALSE	TRUE	FALSE	FALSE
FALSE	FALSE	FALSE	FALSE	TRUE	FALSE	FALSE	FALSE
FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
FALSE	FALSE	FALSE	FALSE	TRUE	FALSE	FALSE	FALSE
FALSE	FALSE	TRUE	FALSE	FALSE	FALSE	FALSE	FALSE
.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.

Table 6.2 Belief states values mapped to trend change indicator values

The obtained dataset is analyzed in the weka, a machine learning tool is used. The dataset is divided into 66% used as training set and 34% as test set. The result of classifying the dataset using J48 pruned tree is summarized below. Correctly classified instances count is 483 which are 97.9716 % and incorrectly classified instances count 10 which are 2.0284 %. This result is shown in the confusion matrix below in the figure 6.5

Figure 6.5 Dataset is validated in weka using J48 decision tree classifier

6.2.4 POMDP Problem Specification

In this section, we discuss about the problem specification of software customization domain in a POMDP file in the format of Tony Cassandra [7] and the dialogue specification parser [6] which is developed by Trung H. Bui, Dennis Hofs and Boris van Schooten at the Human Media Interaction research group of the University of Twente are used. Tony Cassandra file is the input POMDP file format which can be processed by the POMDP solver. It's the formal problem specification file which encoded the domain problem under the defined syntax and semantics. Tony Cassandra POMDP specification file must start 5 lines which specify the discount value, states, actions and observations at the beginning. The Fig 6.1 shows starting 5 lines definition. The order can be in any sequences and all of them must precede specifications of transition probabilities, observation probabilities and rewards. The transition possibilities can be specified in the following format:

T: <action> : <start-state> : <end-state> %f

and observation probabilities are specified in a little similar way with transition probabilities in following format:

O : <action> : <end-state> : <observation> %f

The reward model are specified in this

R: <action> : <start-state> : <end-state> :

<observation> %f format. For any of the entries appeared in the above, an asterisk * for either < state >, < action >, < observation > indicates a wildcard which means this item will be expanded to all existing entities.

discount: %f

values: [reward, cost]

states: [%d, <list of states>]

actions: [%d, <list of actions>]

observations: [%d, <list of observations>]

Figure 6.11: Tony Cassandra POMDP Specification File Starting Line

In the simulated situation software customization, the POMDP file format is designed based on the experiences and domain knowledge. There are 8 machine states specified and user may take 7 kinds of actions which are observations to the machine. The system can perform 13 types of actions. The discount value is 0.95 in this experiment. With this entire specification POMDP file is created.

The POMDP solver adopted in this experiment is 'pomdp-solve'[48]. This program solves problems that are formulated as partially observable Markov decision processes, a.k.a. POMDPs. It uses the basic dynamic programming approach for all algorithms, solving one stage at a time working backwards in time. It does finite horizon problems with or without discounting. It will stop solving if the answer is within a tolerable range of the infinite horizon answer, and there are a couple of different stopping conditions (requires a discount factor less than 1.0). Alternatively you can solve a finite horizon problem for some fixed horizon length. For our experiment, among the implemented POMDP solution algorithms, we have used Two Pass which is selectable with command line options. This POMDP solver is used to generate the alpha file. This alpha file has the
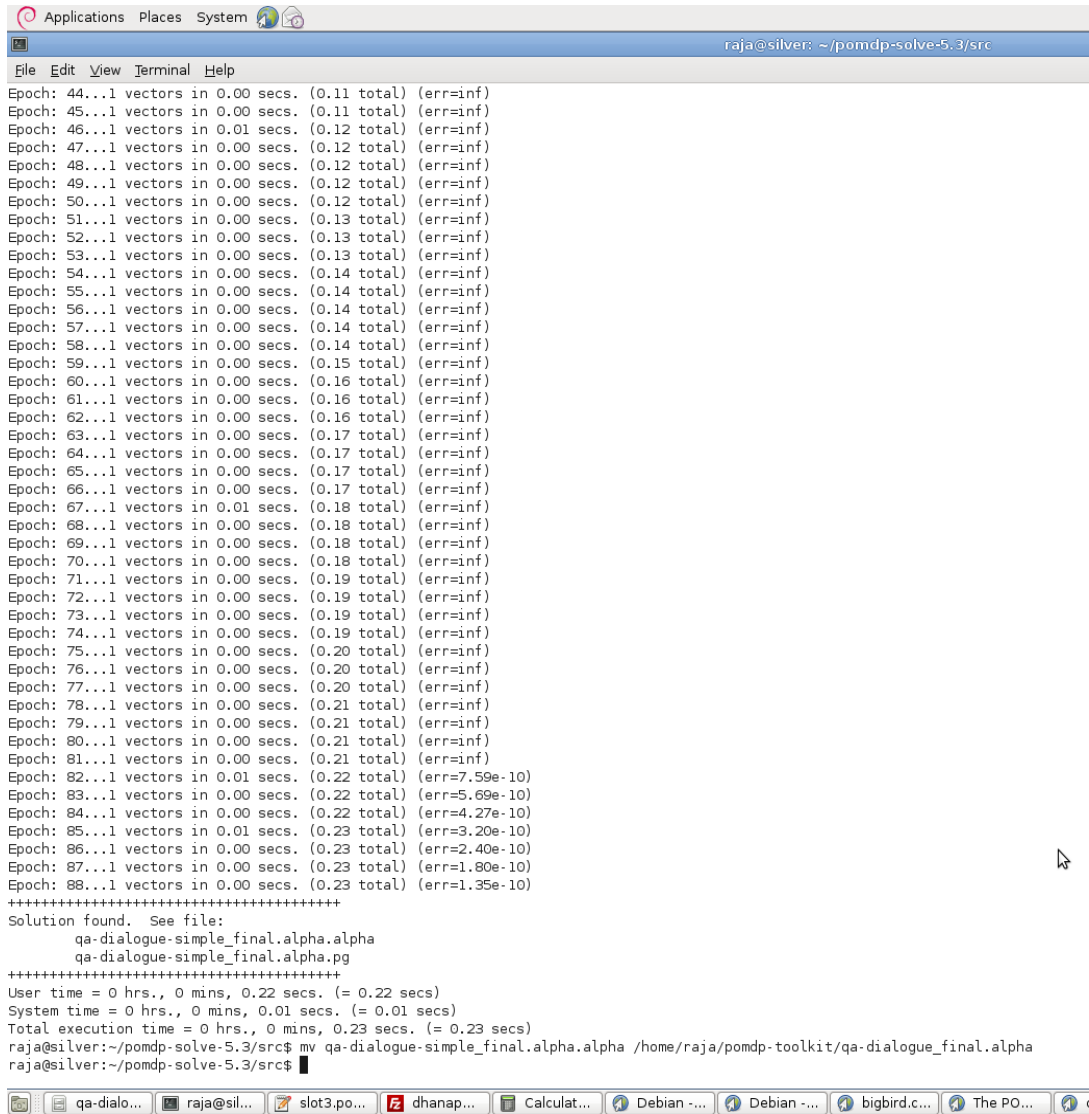
solved policy which POMDP model will used by POMDP model. POMDP solver is
installed in Debian Linux environment.

Figure 6.6 Command line usage of POMDP solver

Figure 6.7 alpha file generation using POMDP solver

Figure 6.8 one of the POMDP files used

The above mentioned POMDP file is used to generate the alpha file which is shown

below in the figure 6.9.

Figure 6.9 Alpha file generated by POMDP solver

Thus, POMDP solver is used in the command line with specifying the time limit to run every epoch with -time_limit. -pomdp to pass the argument of the POMDP input file to generate the alpha file. POMDP solver took 88 turns to generate the alpha file in total taking 0.22 secs, writing the alpha with the POMDP file in the location mentioned with the option -o in the command line execution. There are 4 POMDP files generated with different values in the specification in the same above mentioned procedure. One set of POMDP file along with corresponding alpha file is used one particular mode of operation as mentioned in the proposed work [Strategic, tactical, opportunistic, and scrambled].

6.2.5 ROC of User Intention with Belief Trend Analysis

This module has the implementation of the applying the A-Priori algorithm on the belief data history of all the users used the application. The frequent patterns of all the belief history are compared to the current pattern of belief. The next pattern to the maximum frequency is predicted with confidence.

This section also discusses about the implementation of the analysis of trend in the belief history generated in the previous module. Change point analysis discussed in the proposed work section 5.3 is the technique used to find the points that has change in the trend of the whole history. Finding the points with confidences, ROC of change in trend in the whole history of belief space of the current user and the proposed algorithm is applied. Depending upon the ROC of belief trend the user's intention will be evaluated in each turn. One of the four modes in COCOM is chosen to select the corresponding policy. Action is chosen against the environment by the POMDP model. The action is parser and sent to the ontology integrated module to customize the chosen software.
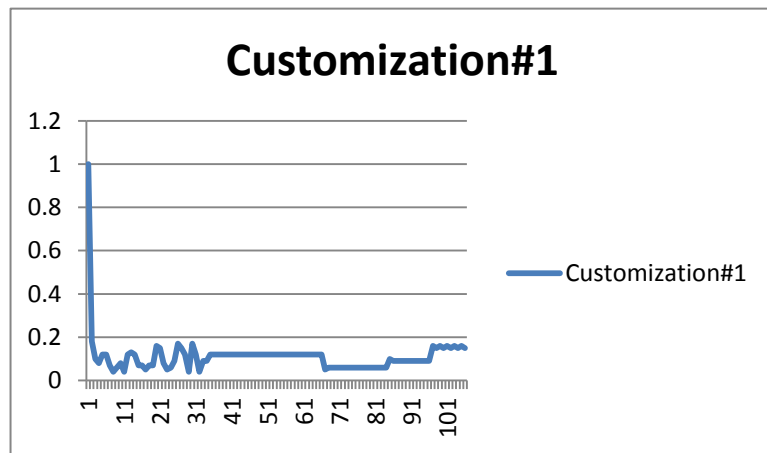
Figure 6.10 sample belief state value graph

The below figure 6.11 shows the graphical user interface of the proposed dialogue system. This has the observations in the left of the panel. In the middle the user-agent interaction history are shown. In the right of the interface window, belief state graph is show which updates for every iteration between user and the agent.

Figure 6.11 Interface of the proposed dialogue system

Figure 6.12 Belief data trend graph

6.3 Experiments and Results

During the dialogue, the agent may take 12 different actions labeled as for example hello, SelectLocateBook, SelectListOfReferences, SelectGetBookContents etc. for conversation in the natural language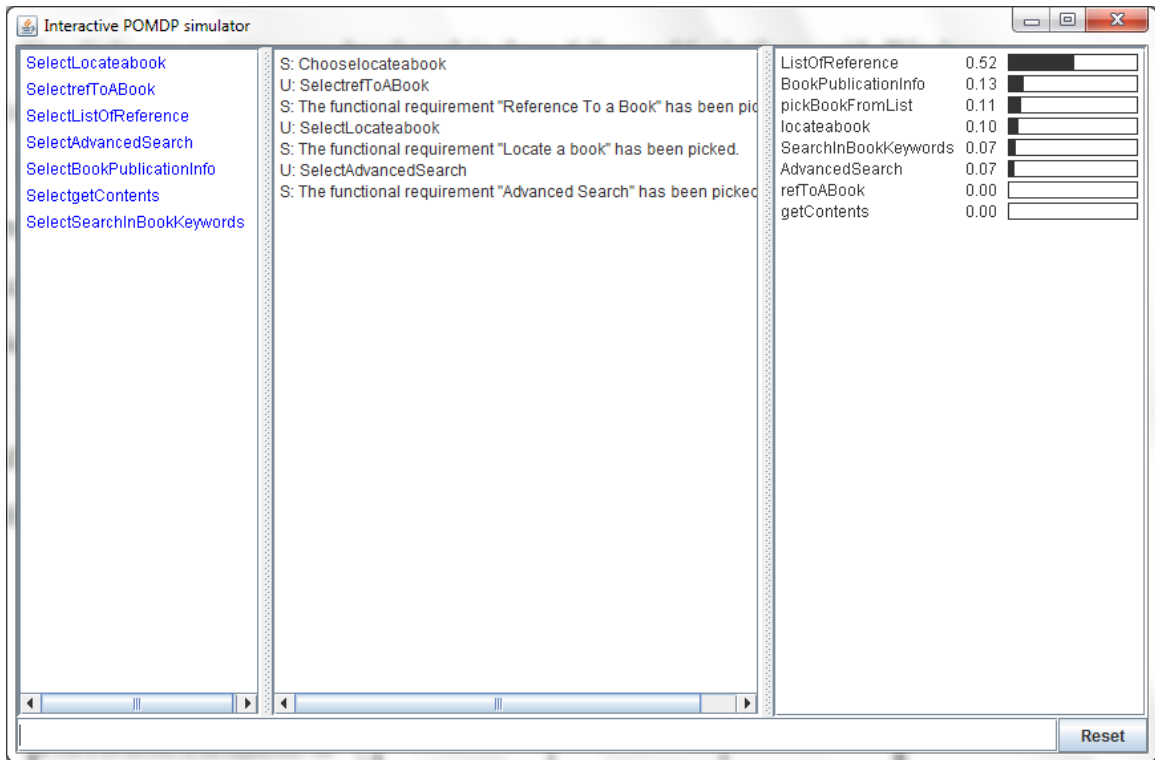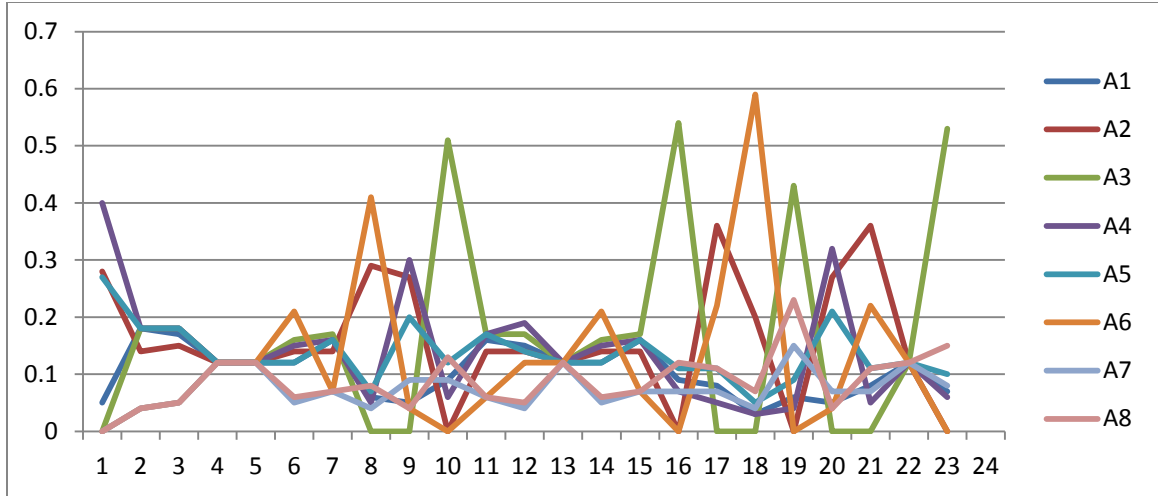. These actions are taken by the machine against the environment based on the observation by the machine from the user. Every software requirement is customized with a goal in the user's mind and this is called user's intention. At each selection of software requirement, belief of the user on a particular package is updated i.e. the probability of choosing the package requirement will be increased and finally. For the purpose of testing, three different scenarios are used to examine the performance of the original and the modified POMDP-based approach with trend analysis and prediction thereby choosing the mode that treats the user in cases with or without change in user intention.

To compare the experimental results with the previous and existing approaches of POMDP, we have used a toolkit known as POMDP Toolkit developed by Bui in 2007 to

carry out experiments and analyze the results of the POMDP dialogue manager [Tool07]. Then we have to parse the specification file to generate a canonical POMDP file in Tony Cassandra's format. We also have installed a solver in order to create an alpha and policy graph file. The solvers mentioned in the toolkit are ZMDP and Perseus. In these solvers, we have create alpha and policy graph files separately. For time consuming and accuracy, we used a different solver known as pomdp-solver, as it creates both the alpha and policy graph files in single execution. We have used this policy graph file for comparing our results with existing POMDP based dialogue management systems.

We have tested our proposed system using software requirement customization domain. We have done several experiments to test our system on considering five cases. 1) The user picks up a goal and achieves it with valid input in both traditional based POMDP dialogue management system and with proposed system. 2) The user picks up a goal and changes it on different stages before achieving it. This is demonstrated in case 2, 3 and 4. 3) The user continues with the previous scenario and achieves the user goal again using valid inputs to the agent with domain knowledge. The detailed description of the cases and the results are described below. The discussion on the result is done in the section 6.4

Case 1: In the first scenario, the user picks up a goal say choosing the software customization #4 with three modules "Locate a book", "Reference to a book" and "List of Relevant Books". This package is a valid one and the user in this scenario is assumed to know this information. This requirement is as shown in the figure 6.3 software customization #4.
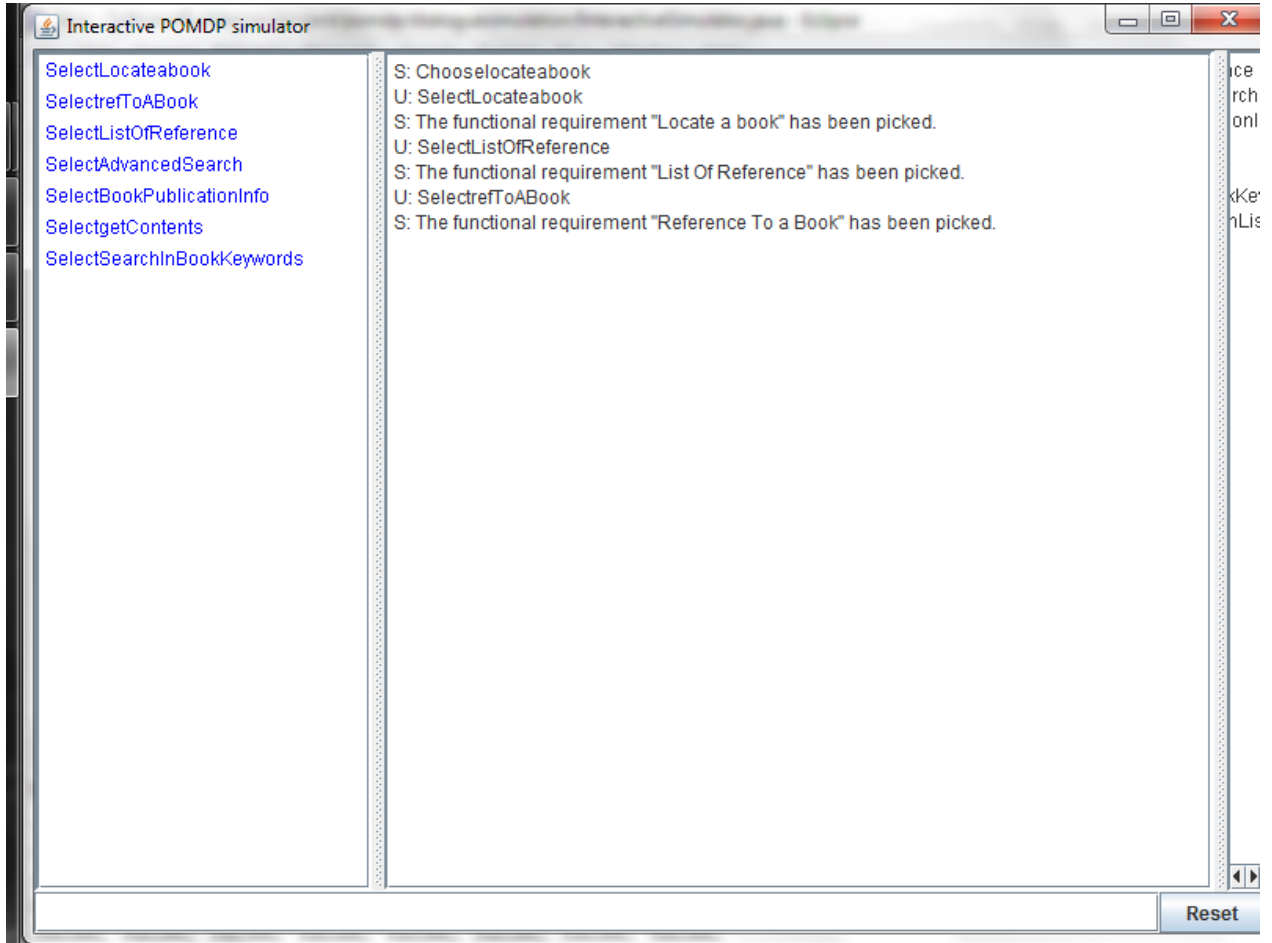
Figure 6.13 User interface Scenario #1

```
Problems  @ Javadoc  Declaration  Console ⊠  History  Error Log
pomdp_dialogue_mgt [Java Application] C:\Program Files\Java\jre6\bin\javaw.exe (Apr 18, 2012 11:38:59 PM)
Mode1
The functional requirement "Locate a book" has been picked.
Locate a book
false, false, false, false, false, false, false, false,
false, false, false, false, false, false, false, false,
false, true, true, true, false, false, false, false,
aPlusCount5,aMinusCount0,bPlusCount3,aCount0
Mode1
The functional requirement "Reference To a Book" has been picked.
Reference To a Book
false, false, false, false, false, false, false, false,
false, false, false, false, false, false, false, false,
false, true, true, true, false, false, false, false,
false, true, true, true, false, false, false, false,
false, true, true, true, false, false, false, false,
false, true, true, false, false, true, false, false,
aPlusCount4,aMinusCount0,bPlusCount4,aCount0
Mode1
The functional requirement "List Of Reference" has been picked.
List Of Reference
```

Figure 6.14 Console output for scenario #1

In the above console output it is shown that the dialogue manager stays at strategic mode for the whole process as the user's intention has not been changed. True and false represents the trend change status of user's goal with the corresponding attribute.

Case 2: In the second scenario, the user picks up a goal of choosing the software customization #3 with three modules "Advanced Search", "List of Relevant Books" and "List of book references". The user changes the goal after choosing "Advanced Search" and "List of Relevant Books" in the software customization #3 to Software requirement customization #6 "Get publication info", "Reference to a book", "Book publication info". The change in the user goal shows significant change in the user's intention and the belief

trend analysis finds the change in the intention and change the mode from strategic to tactical as shown below.
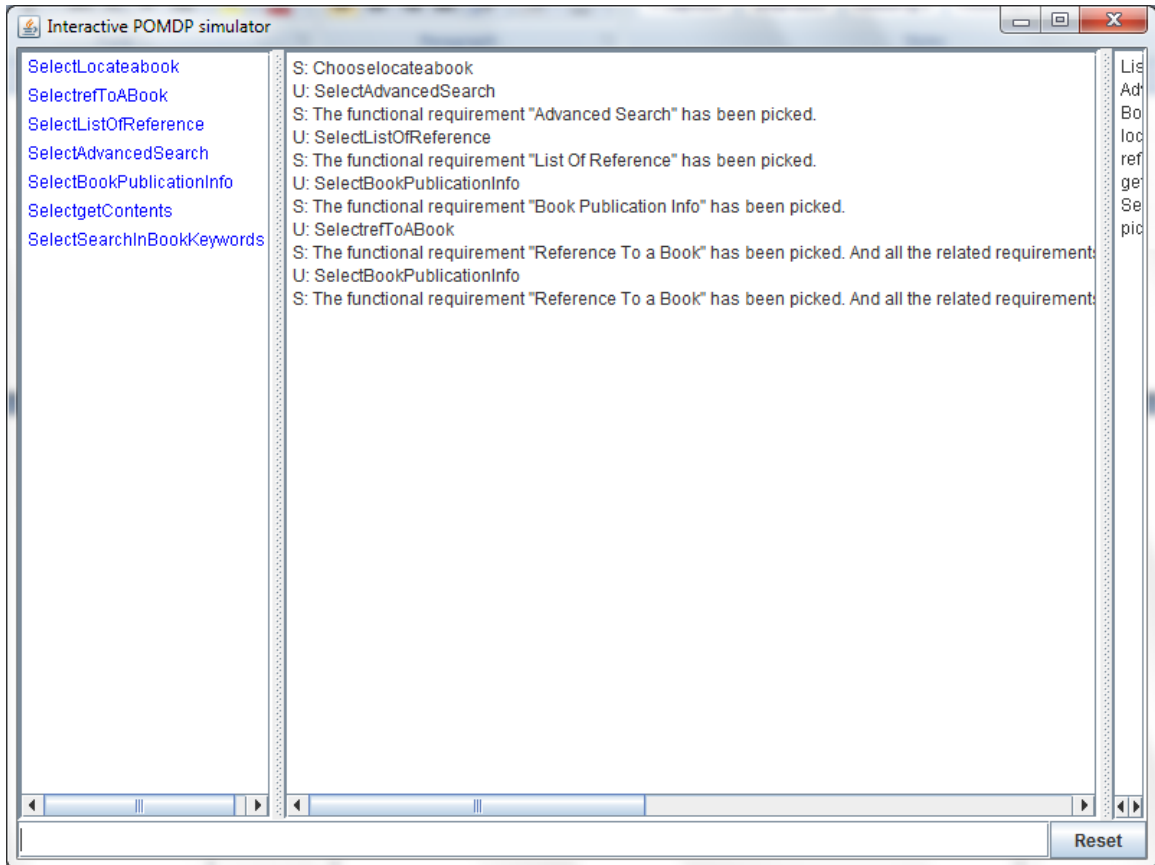

Figure 6.15 User interface Scenario #2
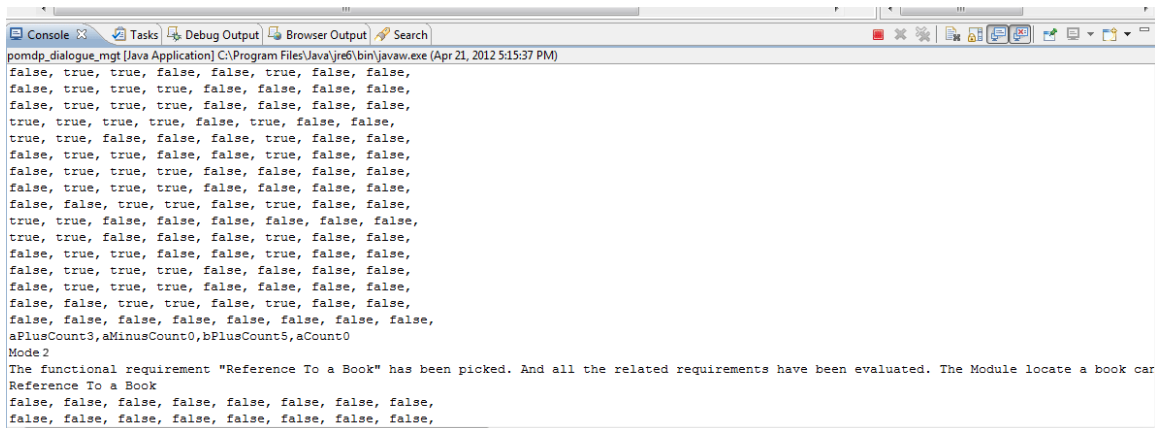
Figure 6.16 Console output for scenario #2
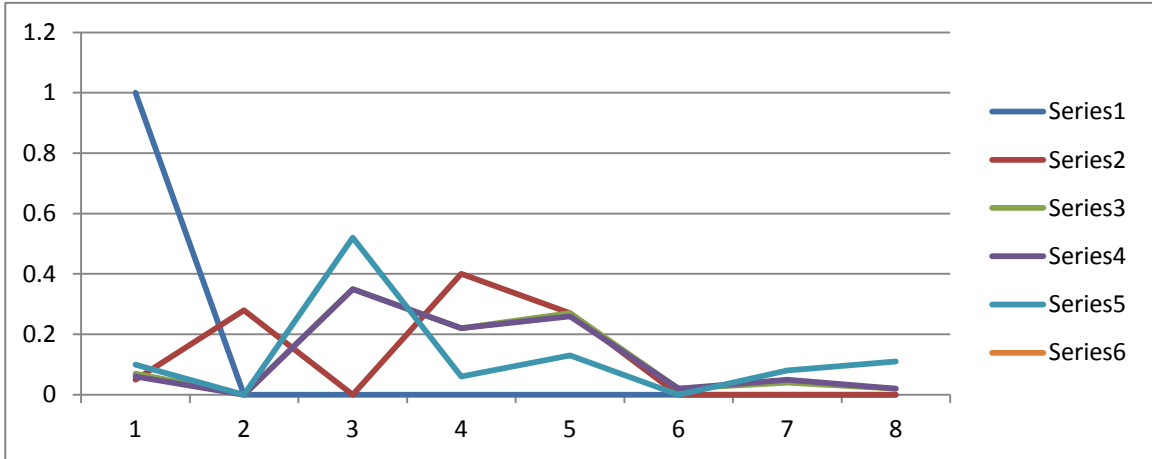
Figure 6.17 Belief data trend graph scenario #2

The above graph shows the belief values plotted for the current user involved in the scenario #2.

Case 3: In the scenario #3, the user picks the goal of customizing requirement #3 and selects locate a book then it selects select reference to a book in requirement #3 after choosing 2 modules the user changes goal to requirement #4. Finally the user changes the user goal for the 3rd time in the conversation choosing requirement #5.
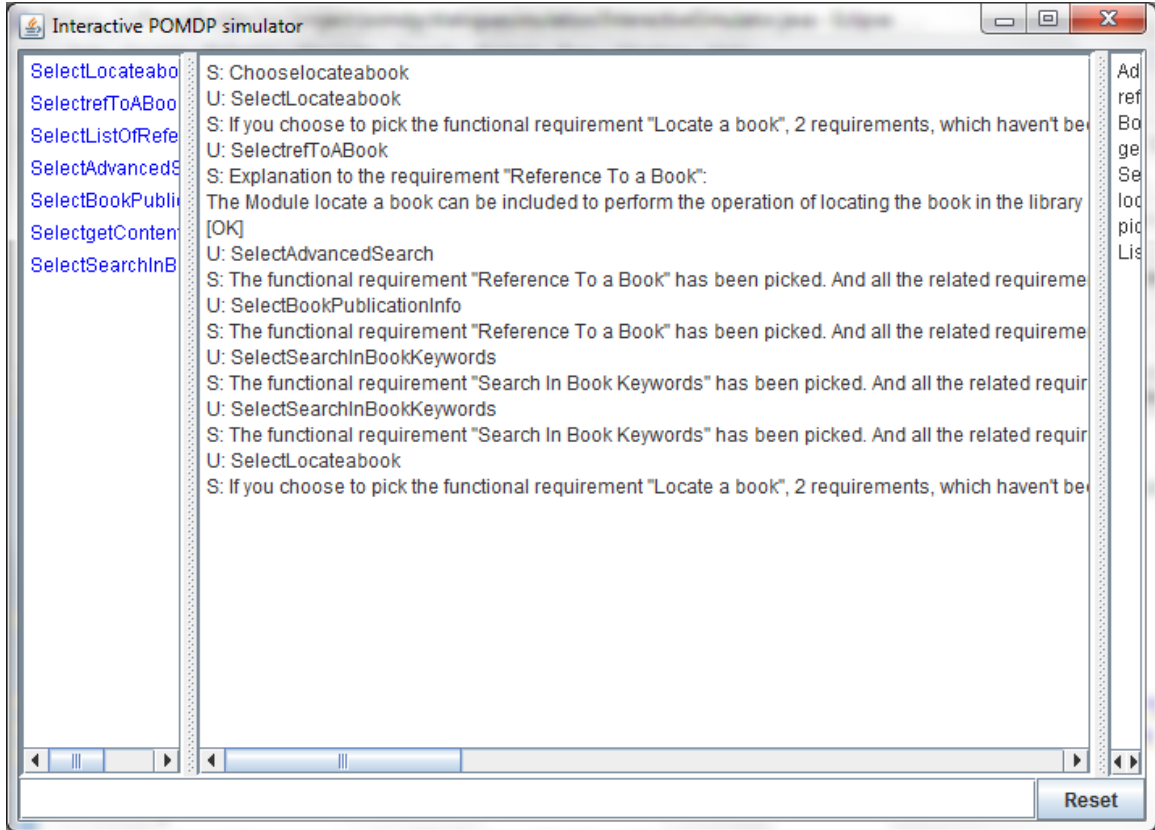
Figure 6.18 User interface Scenario #3
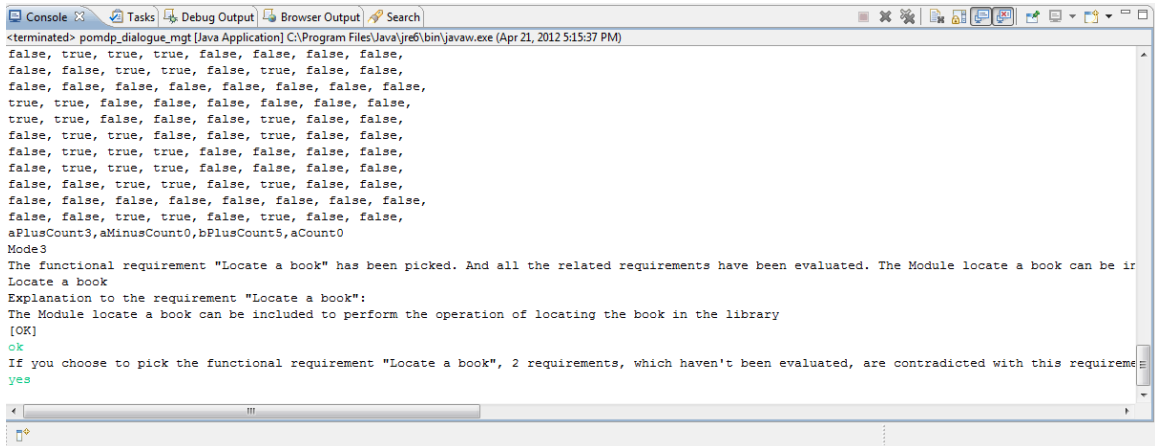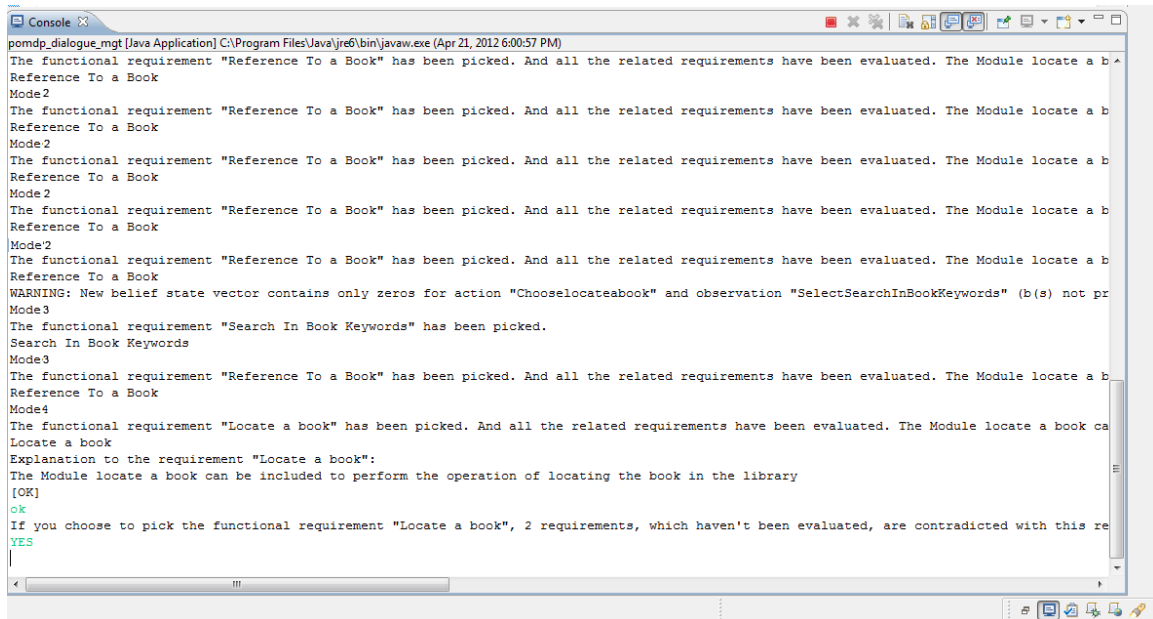
Figure 6.19 Output Console Scenario #3

Chapter 6 Experiments and Discussions

Case 4: In this scenario, the user continues from the case 3 and changes the user goal from requirement #5 and to requirement #4 choosing reference to a book and locate a book modules.

Figure 6.20 Output Console Scenario #4
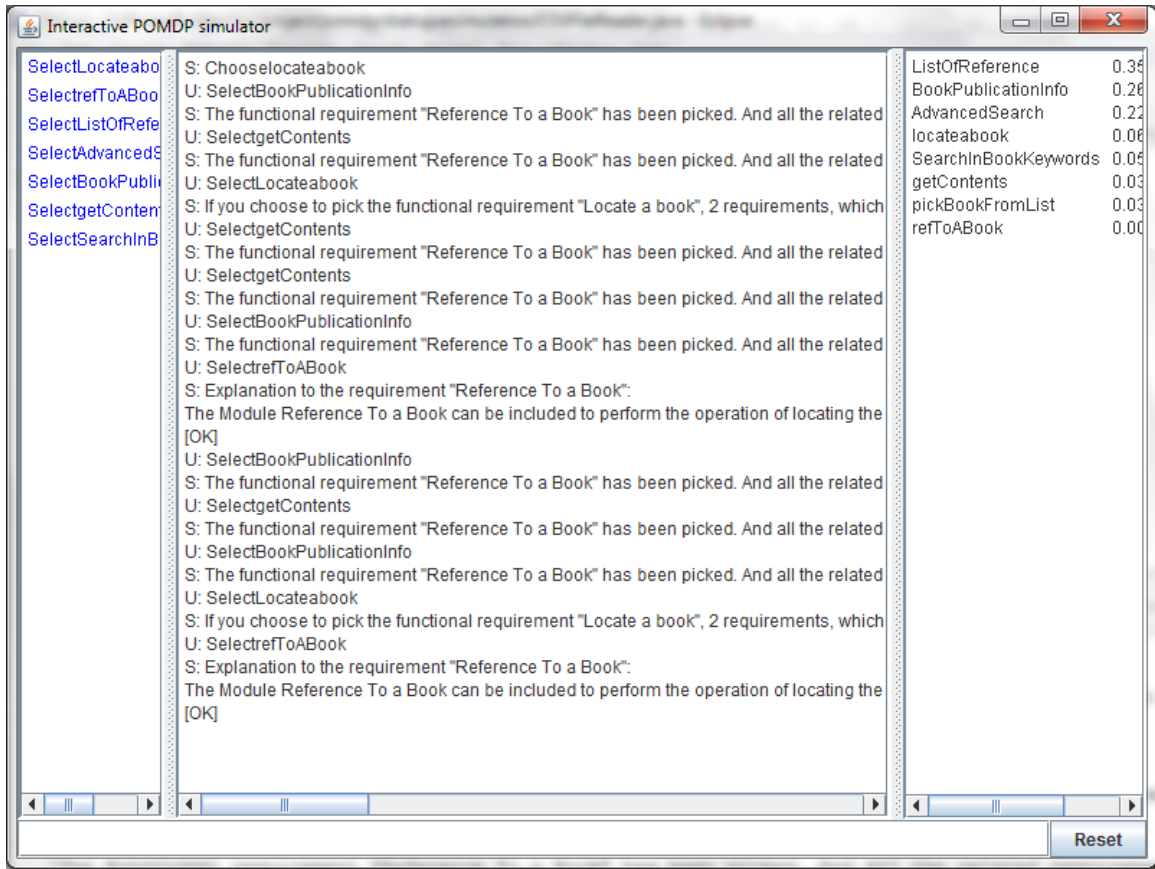
Figure 6.21 User interface Scenario #4

Case#5: this scenario is continued from the previous case. The user continues to choose the requirement #4 in the case 4 and complete the customization as defined in the domain knowledge. The user goal changed till this point is observed to significantly more. The test case results are discussion in the next section.
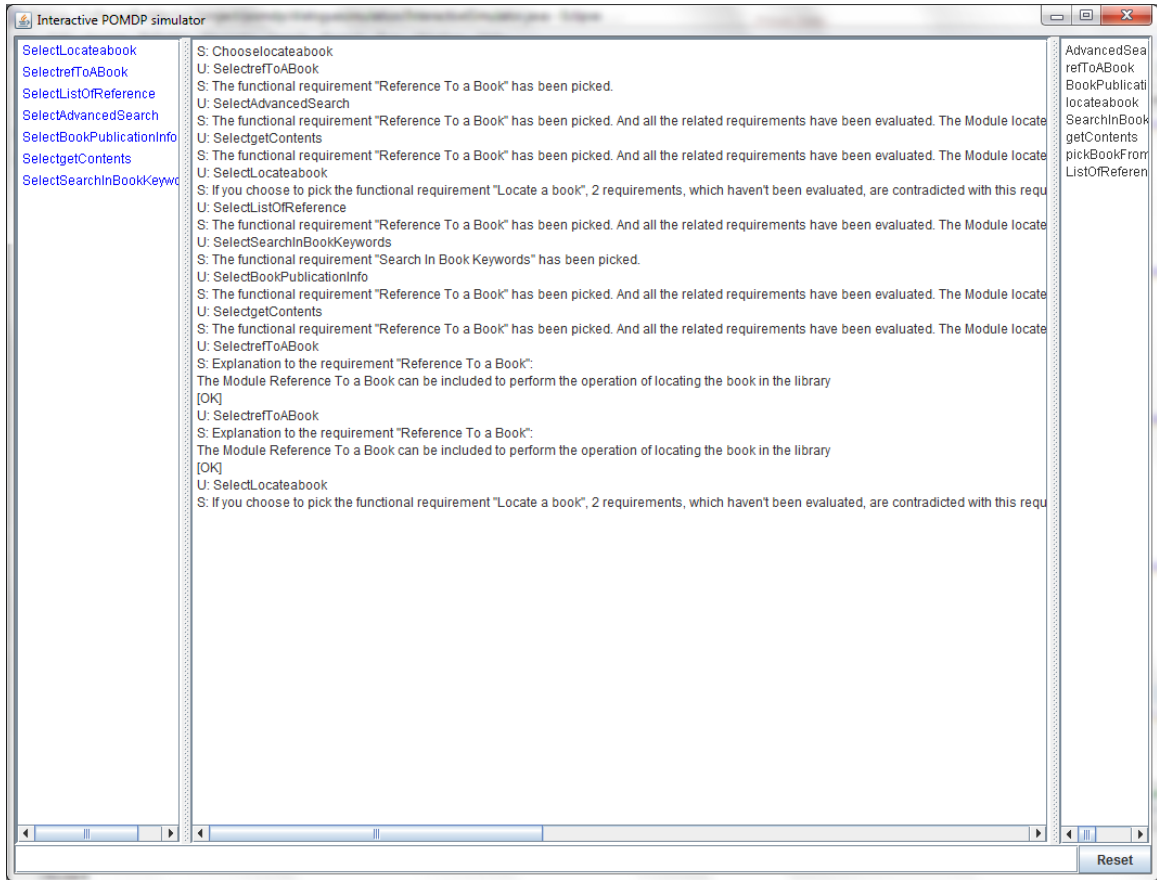
Figure 6.22 User interface Scenario #5
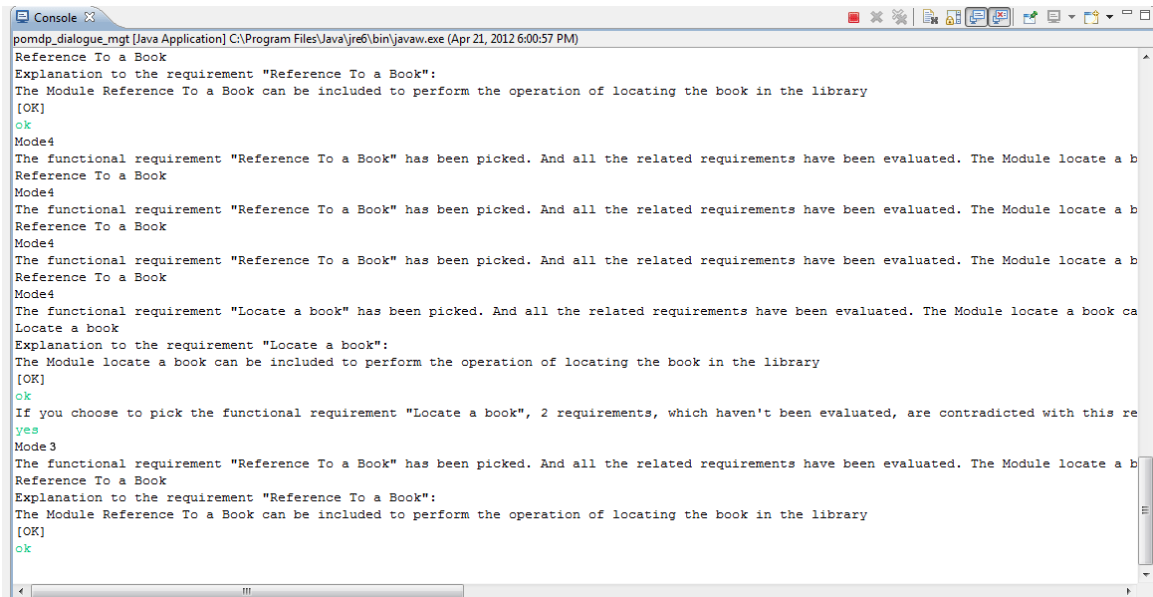
Figure 6.23 Output console Scenario #5

6.4 Discussion

Case 1: A normal dialogue between the user and the system in software customization domain. For each dialogue states, the dialogue manager updates its history and we have collected the belief state values at the end of each dialogue. In the figure 6.3 above, the valid software customization as defined in the ontology is performed. The probability of the user goal increases each stage. The intention of the user which is the belief trend increases steadily. This causes the dialogue management system to stay with strategic mode. The dialogue formed considering the user as knowledge user about the domain and making the conversation straight forward with no question back and forth to confirm their operation. This also avoids giving the user too much of information about the process, Thus saving the time to complete user goal in short period compared to the traditional POMDP-based dialogue management system.

Case 2, 3 and 4: The change of user intention from one valid customization to other results in the drop of belief values in the corresponding attribute. This is detected from the rate of change of trend in the belief values. This causes the mode to switch from strategic to tactical mode as shown in the figure 6.3. When the user keeps selecting the modules which do not exist in one valid customization, the belief values are significantly changed. This causes the agent to switch from tactical to opportunistic and then to scrambled. In tactical mode, every requirement chosen by the user will get explanation about the chosen requirement. In opportunistic mode the user will receive confirmation for all the actions taken to choose any requirement. In scrambled mode the user will receive explanations and confirmation for every dialogue that is conversed with the agent for choosing the requirement.

Case 5: The user continues the dialogue conversation from case 4 and chooses one valid requirements finally. This activity of the user causes the recent belief state of one valid requirement to go higher, which causes the agent to switch from scrambled to opportunistic mode. This recovering back to higher level of contextual control modes provides the dynamic control of the dialogue management system. The traditional POMDP-based dialogue management system doesn't differentiate the user group based on their intention on the attributes of the domain.

Case	Traditional POMDP DM	Libian, 2010	Sabiha, 2010	Proposed System
Avg. number of Unique Dialogue generated for 1000 run	4	6	8	13
Average number of time user achieved the goal per 1000 run	712	721	749	883
Accuracy (Percentage)	71	72	74	88
Standard Deviation	5.6	4.8	4.3	3.1

Figure 6.24: Results comparison of proposed and previous POMDP-based DM.

The above tables shows the results obtained using the simulator for interacting with the proposed system and 3 other POMDP-based dialogue management. The average number of unique dialogues used for 1000 run is shown in the table. The average number of unique dialogues used is higher than the other system which shows that the user is treated with different set of dialogues to get the task done which symbolizes more dynamic control of the strategy. The number of times the user reached goal calculates the accuracy which is more than previous approaches. The results are compared to show that the system works better than the traditional and also with two other systems previously developed.

7. Conclusion and future works

In this thesis, the main dialogue management approaches are reviewed under the computer shopping system. Also the history information space theory is discussed and a thorough analysis of the major approaches of dialogue management approaches with the theory of information space reveals reasons for their problems. With the analysis, the problem of the original POMDP based approach is identified. The Markovian over the belief state in the dialogue management process is problematic because it loses some significant information needed for the decision making. Therefore, the POMDP-based approach applied in the dialogue management cannot detect uncertainties in the belief state which are caused by the change in the user's intention. Change of trend in belief state in the process of planning for the construction of a real truthful, relevant, clear, and informative dialogue system. Based on the theory, a modified approach is proposed to enable POMDP-based dialogue management to handle uncertainties in belief state itself. Experimental results demonstrate significant improvement by the new approach towards accurate recognition of user's intention. Finding the rate of change in the user's intention switches the mode of operation that suits to the user with different level of knowledge about domain. The advantage is more obvious when it comes with the scenario that user has lack of knowledge and provides unreasonable information to the agent. Since when the user is asking for a help, she or he is always lack of the particular domain knowledge, thus the proposed modified approach can be applied to the practical project to provide better services to the human user. The limitations could be the scalability of the POMDP model is still not achieved in this proposed method and the setup is more complicated when comparing to the traditional POMDP as machine learning involves in the planning.

For the future work, active investigation is under way to include the visualization to the POMDP-based dialogue management system with belief trend. Also, another important direction is that to investigate the more practical model to solve the POMDP based approach scale up problem. When the domain is complicated, the states space of POMDP specification file can be really huge and the POMDP solution is computation prohibitive. The current active researches have already put lots of efforts in this area to design more practical framework and POMDP solution algorithm to speed up the approximate solution finding process.

REFERENCES

[1] Trung H. BUI "Multimodal Dialogue Management - State of the art" Human Media Interaction Department University of Twente, 7500 AE Enschede, The Netherlands buith@cs.utwente.nl January 3, 2006 Version 1.0

[2] Young, S. 2000. "Probabilistic Methods in Spoken Dialogue Systems," Philosophical Transactions of the Royal Society (Series A), 358(1769): 1389-1402.

[3] M.F. McTear. Spoken dialogue technology: enabling the conversational user interface. ACM Computing Surveys (CSUR), 34(1): 169, 2002.

[4] TH Bui. Multimodal dialogue management-state of the art. 2006.

[5] PJ Wyard, AD Simons, S. Appleby, E. Kaneen, SH Williams, and KR Preston. Spoken language systems: beyond prompt and response. BT technology journal, 14(1): 187— 205, 1996.

[6] ARETOULAKI,M. AND LUDWIG, B. 1999. Automaton descriptions and theorem-proving: a marriage made in heaven? In Proceedings of IJCAI'99 Workshop on Knowledge and Reasoning in Practical Dialogue Systems (Stockholm, Sweden), IJCAI.

[7] Sta®an Larsson and David Traum. Information state and dialogue management in the TRINDI Dialogue Move Engine Toolkit. In Natural Language Engineering Special Issue on Best Practice in Spoken Language Dialogue Systems Engineering, Cambridge University Press, U.K, pages 323

[8] H. Bunt. Abduction, Belief and Context in Dialogue, volume 1, chapter Dialogue pragmatics and context speci¯cation, pages 81{150. Benjamins, 2000.

[9] W. Xu, B. Xu, T. Huang, and H. Xia. Bridging the gap between dialogue management and dialogue models. In Proceedings of the 3rd SIGdial workshop on Discourse and dialogue-Volume 2, page 210. Association for Computational Linguistics, 2002.

[10] F.V. Jensen and T.D. Nielsen. Bayesian networks and decision graphs. ASA, 2001.

[11] N. Blaylock and J. Allen. A collaborative problem-solving model of dialogue. In 6th SIGdial Workshop on Discourse and Dialogue, 2005.

[12] R.W. Smith and D.R. Hipp. Spoken natural language dialog systems: a practical approach. Oxford University Press, USA, 1994.

[13] J. Searle and S. Acts. An essay in the philosophy of language. Cambridge: CUP, 1969.

[14] W. Xu, B. Xu, T. Huang, and H. Xia. Bridging the gap between dialogue management and dialogue models. In Proceedings of the 3rd SIGdial workshop on Discourse and dialogue-Volume 2, page 210. Association for Computational Linguistics, 2002.

[15] Traum, David R. Larsson, Staffan Kuppevelt, Jan "The Information State Approach to Dialogue Management" 2003 Springer Netherlands Isbn: 978-94-010-0019-2 325-353 Volume: 22 http://dx.doi.org/10.1007/978-94-010-0019-2_15 Doi: 10.1007/978-94-010-0019-2_15

[16] Staffan Larsson, Peter Ljunglof, Robin Cooper, Elisabet EngdahL and Stina Ericsson. GoDiS - An Accommodating Dialogue System. In Proceedings of ANLP/NAACL-2000 Workshop on Conversational Systems, pages 7-10, 2000.

[17] Staffan Larsson and David Traum. Information state and dialogue management in the TRINDI Dialogue Move Engine Toolkit. In Natural Language Engineering Special

Issue on Best Practice in Spoken Language Dialogue Systems Engineering, Cambridge University Press, U.K, pages 323-340, 2000.

[18] Michael Johnston, Srinivas Bangalore, Gunaranjan Vasireddy, Amanda Stent, Patrick Ehlen, Marilyn A. Walker, Steve Whittaker, and Preetam Maloor. Match: An architecture for multimodal dialogue systems. In ACL, pages 376-383, 2002.

[19] D. Hofs, R. op den Akker, and A. Nijholt. A generic architecture and Dialogue model for multimodal interaction. In Proceedings 1st Nordic Symposium on Multimodal Communication, 2003.

[20] D. Traum and S. Larsson. The information state approach to dialogue management. Current and New Directions in Discourse and Dialogue, pages 325-353, 2003.

[21] S. Singh, D. Litman, and M. Kearns. Optimizing dialogue management with reinforcement learning: Experiments with the njfun system. Journal of Artifcial Intelligence Research, 2002.

[22] C. Wai, H.M. Meng, and R. Pieraccini. Scalability and portability of a belief network based dialog model for different application domains. In Proceedings of the first international conference on Human language technology research, page 6. Association for Computational Linguistics, 2001.

[23] T. Paek and E. Horvitz. Conversation as action under uncertainty. In Proceedings of the 16th Conference on Uncertainty in Artificial Intelligence, pages 455—464. Morgan Kaufmann, 2000.

[24] J.D. Williams, P. Poupart, and S. Young. Factored partially observable Markov decision processes for dialogue management. In 4th Workshop on Knowledge and Reasoning in Practical Dialog Systems, pages 76-82, 2005.

[25] E. Levin, R. Pieraccini, and W. Eckert. A stochastic model of human-machine interaction for learning dialog strategies. IEEE Transactions on Speech and Audio Processing, 8(1): 11, 2000.

[26] S.P. Singh, D.J. Litman, M.J. Kearns, and M.A. Walker. Optimizing dialogue management with reinforcement learning: Experiments with the NJFun system. Journal of Artificial Intelligence Research, 16(1): 105-133, 2002.

[27] N. Roy, J. Pineau, and S. Thrun. Spoken dialogue management using probabilistic reasoning. In Proceedings of the 38th Annual Meeting on Association for Computational Linguistics, pages 93-100. Association for Computational Linguistics, 2000.

[28] Varges, S.; Riccardi, G.; Quarteroni, S.; Ivanov, A.V.; , "POMDP concept policies and task structures for hybrid dialog management," Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on , vol., no., pp.5592-5595, 22-27 May 2011 doi:10.1109/ICASSP.2011.5947627

[29]] T. Bui, J. Zwiers, M. Poel, and A. Nijholt. Affective dialogue management using factored POMDPs. Interactive Collaborative Information Systems, pages 207-236, 2010.

[30] S. Thrun. Probabilistic robotics. Communications of the ACM, 45(3):57, 2002

[31] J.D. Williams, P. Poupart, and S. Young. Factored partially observable Markov decision processes for dialogue management. In 4th Workshop on Knowledge and Reasoning in Practical Dialog Systems, pages 76-82, 2005.

[32] Jason D. Williams, Pascal Poupart, and Steve Young. Factored Partially Observable Markov Decision Processes for Dialogue Management. Proc Workshop on Knowledge and Reasoning in Practical Dialog Systems, International Joint Conference on Artificial Intelligence (IJCAI), Edinburgh, 2005.

[33] S. Young. Using POMDPs for Dialog Management. IEEE/ACL Workshop on Spoken Language Technology, Aruba, 2006.

[34] Predicting user mental states in spoken dialogue systems Zoraida Callejas, David Griol and Ramón López-Cózar EURASIP Journal on Advances in Signal Processing 2011, 2011:6 doi:10.1186/1687-6180-2011-6

[35] Gary Klein and David Klinger, "Naturalistic Decision Making" Human Systems IAC GATEWAY Volume XI: Number 3, winter 1991

[36] N A Stanton, M J Ashleigh, A D Roberts, F Xu. Testing Hollnagel's Contextual Control Model: Assessing team behavior in a human Supervisory control task. International Journal of Cognitive Ergonomics, 2001.

[37] Feigh K., Pritchett A. Design of Support Systems for Dynamic Decision Making in Airline Operations. Sch. of Ind. & Syst. Eng., Georgia Inst, of Technology, Atlanta, GA, 2006.

[38] Feigh, Karen and Pritchett, Amy. Design of Multi-mode Support Systems 73 for Airline Operations. International Conference on Human Computer Interaction in Aviation. Eurisco, September 2006.

[39] Jason D. Williams and Steve Young Cambridge "Scaling Up POMDPs for Dialog Management: The Summary POMDP Method" University Engineering Department IEEE 2005.

[40] Hongxing He, Jie Chen, Huidong Jin, Shuheng Chen "Stock Trend Analysis and Trading Strategy" AI-ECON Research Center 2007

[41] Victor Sperandeo 1-2-3 Trend Change http://thepatternsite.com/123tc.html

[42] S. M LaValle. Planning algorithms. Cambridge Univ Pr, 2006.

[43] Taylor, Wayne (2000b), "A Pattern Test for Distinguishing Between Autoregressive and Mean-Shift Data," submitted to Journal of Quality Technologies.

[44] R. Agrawal, T. Imielinski, and A.N. Swami. Mining association rules between sets of items in large databases. In P. Buneman and S. Jajodia, editors, Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data, volume 22(2) of SIGMOD Record, pages 207–216. ACM Press, 1993.

[45] H. Mannila, H. Toivonen, and A.I. Verkamo. Efficient algorithms fordiscovering association rules. In U.M. Fayyad and R. Uthurusamy, editors, Proceedings of the AAAI Workshop on Knowledge Discovery in Databases, pages 181–192. AAAI Press, 1994.

[46] Trung H. Bui, Dennis Hofs, and Boris van Schooten. Dialogue specification parser. Http://wwwhoms. cs .utwente .nl/"hofs/porndp/ , May 2010.

[47] Tony Cassandra. Tony's POMDP Page, http://www.cassandra.org/pomdp/ index.shtml , May 2010.

[48] POMDP's home page by The Cassandra Organization http://www.pomdp.org/ pomdp/code/index.shtml

[49] Xieshen Zhang "An Interactive Approach of Ontology-based Requirement Elicitati-on for Software Customization "A Thesis, University of Windsor, Canada. 2011

[50] X Wei and AI Rudnicky. 1999. An agenda-based dialog management architecture for spoken language systems. In Proc. of IEEE ASRU. Seattle, WA.

[51] Gupta, Ramachandran, Deepak, Raux, Antoine, Mehta, Neville, Krawczyk, Stefan Hoffman, Matthew   2012   BELIEF TRACKING AND ACTION SELECTION IN SPOKEN DIALOG SYSTEMS   United States HONDA   MOTOR   CO.,   LTD. (Tokyo, JP)   20120053945 http://www.freepatentsonline.com/y2012/0053945.html

[52] CHAP, Lin, Yong,Makedon, Fillia, Gulrez, Tauseef, Hassanien, Aboul Ella, From Dialogue Management to Pervasive Interaction Based Assistive Technology Advances in Robotics and Virtual Reality Intelligent Systems Reference Library 2012 Springer Berlin Heidelberg - 978-3-642-23363-0 - Engineering http://dx.doi.org/10.1007/978-3-642-23363-0_8- 10.1007/978-3-642-23363-0_8

[53] Williams, Jason (New York, NY, US)   2012   User intention based on N-best list of recognition hypotheses for utterances in a dialog   United States   AT&T   Intellectual Property I, L.P. (Atlanta, GA, US)   8140328

[54] CHAP, riol, David, Carbó, Javier, Molina, José, Novais, Paulo, Hallenborg, Kasper, Tapia, Dante I.,Rodríguez, Juan M. Corchado, Optimizing Dialog Strategies for Conversational Agents Interacting in AmI Environments Ambient Intelligence - Software and Applications, Advances in Intelligent and Soft Computing 2012 PB - Springer Berlin / Heidelberg SN - 978-3-642-28782-4.

[55] Dongho Kim; Kim, J.H.; Kee-Eung Kim; , "Robust Performance Evaluation of POMDP-Based Dialogue Systems," Audio, Speech, and Language Processing, IEEE Transactions   on ,   vol.19,   no.4,   pp.1029-1040,   May   2011 doi: 10.1109/TASL.2010.2076394

[56] Zhou, Weidong; Yuan, Baozong; , "Trainbot: A spoken dialog sytem using partially observable Markov Decision Processes," Wireless, Mobile and Multimedia Networks (ICWMNN 2010), IET 3rd International Conference on , vol., no., pp.381-384, 26-29 Sept.2010 doi: 10.1049/cp.2010.0695

[57] Jing-Doo Wang; , "A novel approach to compute pattern history for trend analysis," Fuzzy Systems and Knowledge Discovery (FSKD), 2011 Eighth International Conference on , vol.3, no., pp.1746-1750, 26-28 July 2011 doi: 10.1109/FSKD.2011.6019799

[58] Baranovskyi, O.; , "Content — Monitoring as a new trend in quantitative evaluation of changes in social processes," Intelligent Data Acquisition and Advanced Computing Systems (IDAACS), 2011 IEEE 6th International Conference on , vol.2, no., pp.518-521, 15-17 Sept.2011 doi: 10.1109/IDAACS.2011.6072820

[59] Yongji Fu; Bryan, H.S.; , "Rapid trend detection for an ambulatory monitoring system," Engineering in Medicine and Biology Society,EMBC, 2011 Annual International Conference of the IEEE , vol., no., pp.1177-1180, Aug. 30 2011-Sept. 3 2011 doi: 10.1109/IEMBS.2011.6090276

[60] Taha, T.; Miro, J.V.; Dissanayake, G.; , "POMDP-based long-term user intention prediction for wheelchair navigation," Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on , vol., no., pp.3920-3925, 19-23 May 2008 doi: 10.1109/ROBOT.2008.4543813

[61] Chowdhary, K.R.; Bansal, V.S.; , "Information Retrieval using probability and belief theory," Emerging Trends in Networks and Computer Communications (ETNCC), 2011 International Conference on , vol., no., pp.188-191,22-24 April 2011 cdoi: 10.1109/ETNCC.2011.5958513

[62] Boukhris, I.; Elouedi, Z.; Benferhat, S.; , "Analyzing belief function networks with conditional beliefs," Intelligent Systems Design and Applications (ISDA), 2011 11th International Conference on , vol., no., pp.959-964, 22-24 Nov. 2011.

[63] Cheng Zeng; Liang Hong; Jian Wang; Chuan He; Jilei Tian; Xiaogang Yang; , "Role-Based Contextual Recommendation," Internet of Things (iThings/CPSCom), 2011 International Conference on and 4th International Conference on Cyber, Physical and Social Computing , vol., no., pp.598-601, 19-22 Oct. 2011.

[64] Minoya, K.; Arita, T.; Omori, T.; , "An artificial life approach for investigating the emergence of a Theory of Mind based on a functional model of the brain," Artificial Life (ALIFE), 2011 IEEE Symposium on , vol., no., pp.108-115, 11-15 April 2011

[65] Meguro, T.; Minami, Y.; Higashinaka, R.; Dohsaka, K.; , "Wizard of Oz evaluation of listening-oriented dialogue control using POMDP," Automatic Speech Recognition and Understanding (ASRU), 2011 IEEE Workshop on , vol., no., pp.318-323, 11-15 Dec. 2011

VITA AUCTORIS

Rajaprabhu Dhanapal was born in the year 1986 Tamilnadu, India. He received his Bachelor's of Technology degree with the specialization in Information Technology from Anna University, Tamil Nadu, India. He joined the University of Windsor in January 2011 and is currently a candidate for the Master's degree in Computer Science under the supervision of Dr. Xiaobu Yuan.