

University of Windsor

## Scholarship at UWindor

---

Electronic Theses and Dissertations

Theses, Dissertations, and Major Papers

---

2010

# Four Mode Based Dialogue Management with Modified POMDP Model

Sabiha Sathulla  
*University of Windsor*

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

---

### Recommended Citation

Sathulla, Sabiha, "Four Mode Based Dialogue Management with Modified POMDP Model" (2010).  
*Electronic Theses and Dissertations*. 7914.  
<https://scholar.uwindsor.ca/etd/7914>

This online database contains the full-text of PhD dissertations and Masters' theses of University of Windsor students from 1954 forward. These documents are made available for personal study and research purposes only, in accordance with the Canadian Copyright Act and the Creative Commons license—CC BY-NC-ND (Attribution, Non-Commercial, No Derivative Works). Under this license, works must always be attributed to the copyright holder (original author), cannot be used for any commercial purposes, and may not be altered. Any other use would require the permission of the copyright holder. Students may inquire about withdrawing their dissertation and/or thesis from this database. For additional inquiries, please contact the repository administrator via email ([scholarship@uwindsor.ca](mailto:scholarship@uwindsor.ca)) or by telephone at 519-253-3000ext. 3208.

# **Four Mode Based Dialogue Management with Modified POMDP Model**

**by**

**Sabiha Sathulla**

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

2010

© 2010 Sabiha Sathulla



Library and Archives  
Canada

Published Heritage  
Branch

395 Wellington Street  
Ottawa ON K1A 0N4  
Canada

Bibliothèque et  
Archives Canada

Direction du  
Patrimoine de l'édition

395, rue Wellington  
Ottawa ON K1A 0N4  
Canada

*Your file* *Votre référence*  
ISBN: 978-0-494-70594-0  
*Our file* *Notre référence*  
ISBN: 978-0-494-70594-0

#### NOTICE:

The author has granted a non-exclusive license allowing Library and Archives Canada to reproduce, publish, archive, preserve, conserve, communicate to the public by telecommunication or on the Internet, loan, distribute and sell theses worldwide, for commercial or non-commercial purposes, in microform, paper, electronic and/or any other formats.

The author retains copyright ownership and moral rights in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

#### AVIS:

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque et Archives Canada de reproduire, publier, archiver, sauvegarder, conserver, transmettre au public par télécommunication ou par l'Internet, prêter, distribuer et vendre des thèses partout dans le monde, à des fins commerciales ou autres, sur support microforme, papier, électronique et/ou autres formats.

L'auteur conserve la propriété du droit d'auteur et des droits moraux qui protègent cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

---

In compliance with the Canadian Privacy Act some supporting forms may have been removed from this thesis.

While these forms may be included in the document page count, their removal does not represent any loss of content from the thesis.

Conformément à la loi canadienne sur la protection de la vie privée, quelques formulaires secondaires ont été enlevés de cette thèse.

Bien que ces formulaires aient inclus dans la pagination, il n'y aura aucun contenu manquant.

  
**Canada**

# **Author's Declaration of Originality**

I hereby certify that I am the sole author of this thesis and 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. 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 this thesis has not been submitted for a higher degree to any other University or Institution.

# Abstract

This thesis proposes a method to manage the interaction between the user and the system dynamically, through speech or text input which updates the user goals, select system actions and calculate rewards for each system response at each time-stamp. The main focus is made on the dialog manager, which decides how to continue the dialogue. We have used POMDP technique, as it maintains a belief distribution on the dialogue states based on the observations over the dialogue even in a noisy environment. Four contextual control modes are introduced in dialogue management for decision-making mechanism, and to keep track of machine behaviour for each dialogue state. The result obtained proves that our proposed framework has overcome the limitations of prior POMDP methods, and exactly understands the actual intention of the users within the available time, providing very interactive conversation between the user and the computer.

# **Dedication**

To my Parents and all my Friends

## **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. Dan Wu and Dr. Fritz Rieger, for being in the thesis committee and spending their valuable time in providing me with encouraging feedback.

I am deeply indebted to express my gratitude to my colleagues Libian Bian and Mohit Sud for their moral support and valuable suggestions.

Finally, I would like to thank all my friends who were with me by providing moral support and took care of me when I had my nasal surgery while I was doing my thesis.

# Contents

<b>Abstract</b>	<b>iv</b>
<b>Dedication</b>	<b>v</b>
<b>Acknowledgements</b>	<b>vi</b>
<b>List of Figures</b>	<b>x</b>
<b>List of Tables</b>	<b>xii</b>
<b>1 Introduction and Motivation</b>	<b>1</b>
1.1 Introduction .....	1
1.2 The Origin .....	1
1.3 Overview of Spoken Dialog System .....	3
1.4 Motivation .....	4
1.5 Problem Statement .....	4
1.6 Thesis Structure .....	5
<b>2 Background</b>	<b>6</b>
2.1 Dialogue Management – An Overview .....	6
2.1.1 Input Mode .....	7
2.1.2 Fusion .....	8
2.1.3 Dialogue Manager .....	8
2.1.4 Fission .....	9
2.1.5 Output Mode .....	10
2.1.6 Multimodal Dialogue Management .....	10

2.2	Review of existing Approaches .....	10
2.2.1	Finite state-based Approach .....	11
2.2.2	Frame-based Approach .....	12
2.2.3	Information state-based Approach .....	13
2.2.4	Probabilistic Approach .....	14
2.2.5	Plan-based Approach .....	15
2.2.6	Agent-based Approach .....	16
2.3	Contextual Control Model .....	16
2.3.1	Four Control Modes .....	17
2.4	Conclusion .....	19
<b>3</b>	<b>Literature Review of Dialogue Management techniques</b> .....	<b>20</b>
3.1	Dialogue Management Techniques .....	20
3.1.1	Handcrafted Dialogue Managers .....	20
3.1.2	Bayesian Network for Dialogue Management .....	22
3.1.3	Supervised Learning and Markov Decision Process .....	24
3.1.4	Semi- Markov Decision Process .....	30
3.1.5	Partially Observable Markov Decision Process .....	32
3.1.5.1	Factored POMDP .....	40
3.1.5.2	Hidden Information State model .....	43
3.1.6	Limitations of Existing POMDP Approaches .....	45
3.2	Review of Contextual Control Model .....	45
3.2.1	Testing COCOM by assessing team behaviour .....	45
3.2.2	COCOM for Dynamic Decision Making .....	48

3.3 Conclusion .....	49
<b>4 Four Mode Based Dialogue Management</b>	<b>50</b>
4.1 Method Description .....	50
4.2 COCOM for Decision Making in DM .....	51
4.3 Modified POMDP Approach .....	52
4.3.1 Rewards .....	54
4.3.2 Confidence Scores .....	55
4.4 Conclusion .....	59
<b>5 Evaluation and Results</b>	<b>60</b>
5.1 Implementation .....	60
5.2 Results .....	63
5.3 Discussion .....	67
5.4 Conclusion .....	68
<b>6 Conclusions and Future Work</b>	<b>69</b>
<b>Bibliography</b>	<b>71</b>
<b>Vita Auctoris</b>	<b>79</b>

## List of Figures

2.1	General Architecture of Dialogue Management System .....	7
2.2	General architecture of Dialogue Manager .....	9
3.1	HC1 handcrafted Dialogue Controller .....	21
3.2	Dependency graph for Spoken Dialogue Management .....	23
3.3	Supervised Learning for action selection .....	25
3.4	Sequential decision Process of ATIS .....	27
3.5	Three possible strategies of ATIS Spoken Dialog System .....	28
3.6	A graph of basic MDP underlying the Dialog Manager .....	29
3.7	The simulated environment interaction for Human-Machine conversations .....	31
3.8	Structure of a Spoken Dialog System .....	33
3.9	The agent – Environment Interaction .....	34
3.10	Influence diagram of continuous POMDP .....	37
3.11	Example conversation in Pizza- Ordering domain .....	38
3.12	Frameworks for modeling uncertainty and policy optimisation .....	40
	in Spoken Dialogue Systems	
3.13	HIS Dialogue Manager .....	44
3.14	Internal structure of Contextual Control Model (COCOM) .....	46
3.15	Contextual Control Model by Eric Hollnagel in 1998 .....	48
4.1	Flow-chart of Proposed Model .....	58
5.1	User Interface of Proposed system .....	61

5.2 Dialogue Manager updates all the activities of the system .....	62
5.3 User Interface for case 1 experiment .....	63
5.4 The updated history component of proposed dialogue manager for case 1 .....	64
5.5 User interface for case 2 experiments .....	64
5.6 The updated history component of proposed dialogue manager for case 2 .....	65
5.7 User Interface for case 3 and 4 experiment .....	65
5.8 The updated history component of proposed dialogue manager for case 3 and 4 .....	66

## List of Tables

4.1 Pseudo- code of Proposed Model .....	57
5.1 Experimental Results of Modified POMDP model .....	67

# **Chapter 1**

## **Introduction and Motivation**

### **1.1 Introduction**

This chapter introduces the origin and overview of the thesis. First, we have narrated the origin of the dialogue management systems briefly and the related fields associated with it. Then the subsequent sections we have illustrated an overview with several examples. Then, the next section follows the motivation of this thesis followed by the problem statement. Finally, the chapter ends with a detailed summary of this thesis.

### **1.2 The Origin**

Human- Computer Interaction is concerned with the joint performance of tasks by human and the machines [Oldsig]. The basic goal is to make the computers interact with the human users, guide and help them in various domains. This area explored in 1963 by Ivan Sutherland in Sketchpad, where visible objects on the screen are directly manipulated with a pointing device such as a light-pen. Then, the actual work started with gesture recognition using a light-pen which is considered as the first application in HCI [Brad98]. Even though, many methods and applications in HCI find their application in various products, still some of basic applications are in the state of research since 1980.

HCI has collaboration with different fields of science. Many of its methods and techniques play a significant role in the field of Artificial Intelligence and other related fields. Sometimes, people think HCI as part of AI, as some of the basic learning techniques for HCI

are extracted from AI. In AI, few concepts such as gesture and speech recognition are used for robot navigation. In Multimedia, HCI concepts such as integrated text and graphics are used to make a movie using video and computer graphics. The basic research is done using speech, text or gesture as input to process the HCI and AI based systems. The growth and demand of video games are one such example.

Consequently, lot of research has been done in HCI particularly focussing on the field of natural language understanding and generation. With the growing demand of various applications of HCI, gives evolution to a new technology in 1980s known as dialog system in which text, speech or gestures are used as input to interact with the humans. Even though, lot of prior research has been done in this field, dialogue systems emerged in various applications in 1990s. One such application is in the field of telecommunications. It uses a telephone or text wizard to interact with the human, process and offer services requested by user using natural language processor. Dialogue management consists of several components to understand the request and provides service to the user. Basically, these types of systems are domain- oriented and works according to the background of intelligent systems.

Different applications where dialogue management technology is used are listed below:

- Telephone based system to provide information or support to human users.
- It has a wide application in the field of Robotics such as Robot Navigation System.
- In the field of virtual reality, the users can interact with the computer using sensory information like speech and gestures in the simulated environment to experience the real-world environment.
- In medical care, various applications such as access and assistive systems are developed to help/assist children, elderly and disability people.

- In multimedia and 3-D technology, it has a wide growth in developing video games.

### **1.3 Overview of Spoken Dialogue System**

A Spoken Dialogue system is a computational device or agent that engages in interaction with other human, uses human language in some form such as speech, text, or gesture and typically engages with human such interaction across multiple turns or sentences [Jods]. It uses a discourse generator or computational linguistics to extract the information from the user and provide services according to the request. Several researches have been done to answer the following questions:

1. What information should be extracted to process the user query?
2. How the information is extracted?
3. How the system decides, what response should be given to the user query?
4. Does the system understand the exact intension of the user?

The first two questions are answered using the existing techniques and deals with input components of the dialogue management. Next two questions deals with the dialogue manager which is still under research. The final question is the most challenging part of Dialogue management and it's been taken care of in this thesis.

### **1.4 Motivation**

In the last decade, though lot of systems are developed on the basis of dialogue management, a number of issues arise to be taken care of. With the growth and development of different techniques to solve the prior issues, several limitations also increased which makes the researchers to concentrate more on numerous applications. Though, the applications use natural language processor for understanding user request, it is not appropriate in several

circumstances. It fails in the case that it is not language independent. Due to higher error rate and reliability of these types of systems in real-time, lot of applications failed to handle exact interaction between the human and the computer. Though, it consists of various components, dialog manager is considered as the heart of the dialogue management system as it is responsible for decision making. In order to handle the above situations, researchers developed different techniques to make the dialog manager to make exact decision and give appropriate response to the user. One such technique is Partially Observable Markov Decision Process (POMDP). Though, Pomdp has emerged to overcome the failures of prior techniques, it has its own limitations to be taken care of. This motivates me to conduct a research on this particular technology and solve the challenges associated with Pomdp based dialog manager by introducing a decision making mechanism based on four-mode concept.

## **1.5 Problem Statement**

This thesis addresses the limitations of Pomdp and methods proposed to overcome the limitations by introducing a different decision-making mechanism using Contextual Control Modes which is addressed as four-mode concept in this thesis. We have modelled a dialogue manager for a real time Pizza-ordering system and modified the Pomdp dialog manager in which the application tracks the transition of system states, actions taken by the system to respond accurately to the user, system response time for the user query, confidence score for the dialog and change in mode for making decisions. We have also modelled a new reward model based on the four-mode decision making mechanism.

## **1.6 Thesis Structure**

The rest of the thesis is organized as follows. Chapter 2 gives a brief background about dialogue management systems and its components. Literature review of existing techniques is also discussed in this chapter with examples. Finally, Contextual control model which is the main heart of the thesis is discussed. Chapter 4 describes the proposed model and modified POMDP approach in accordance with four Contextual Control Modes. Experimental results and performance analysis of the proposed method are presented in chapter 5. Finally, we conclude this thesis with some discussion and future work.

## **Chapter 2**

### **Background**

This chapter gives a brief background on different dialog manager. First, the section starts with Dialogue Management and we have also explained its architecture with some potential applications. Then the components of the dialog system are discussed, giving main focus on dialog manager. The main task of dialog manager is decision making. In this thesis, we have used contextual control modes for making decisions according to user's request. Finally, this chapter ends with a brief background on contextual control modes followed by its existing applications.

#### **2.1 Dialogue Management: An Overview**

A dialogue management system is a machine that interacts with the people by understanding their intention/goal and provides the required information services [Infriek]. With the recent advances in Robotics and Human – Computer Interaction, dialogue management has grabbed more attention from the researchers. In Nineteenth century, where the computers had less importance, people used to travel to the shop to get the reviews, choices, merits and demerits of a particular product from the sales persons. But nowadays, computers play a vital role in our day to day life. With the advancement of computer Avatars, we can shop anything from anywhere, talk to a virtual sales person regarding the product on the web sitting in your drawing room. The Virtual sales assistant is nothing but a computer program which acts like a human, understands the user requirement and provides service according to human request. But, the question which arises that, what does it expects the user could possibly intend to do?

Does the machine offer the user a clear picture of its possible actions? And is it clear what the results of these actions are and whether they were successfully performed? Do these machines know what context of interaction is that they are in? [Infriek] These types of questions always rise up among the researchers while designing a virtual avatar. These types of dialogue systems are flexible and practical enough to control a natural human – computer dialogue. They are mostly task-oriented and provide domain portability in order to allow users in various applications.

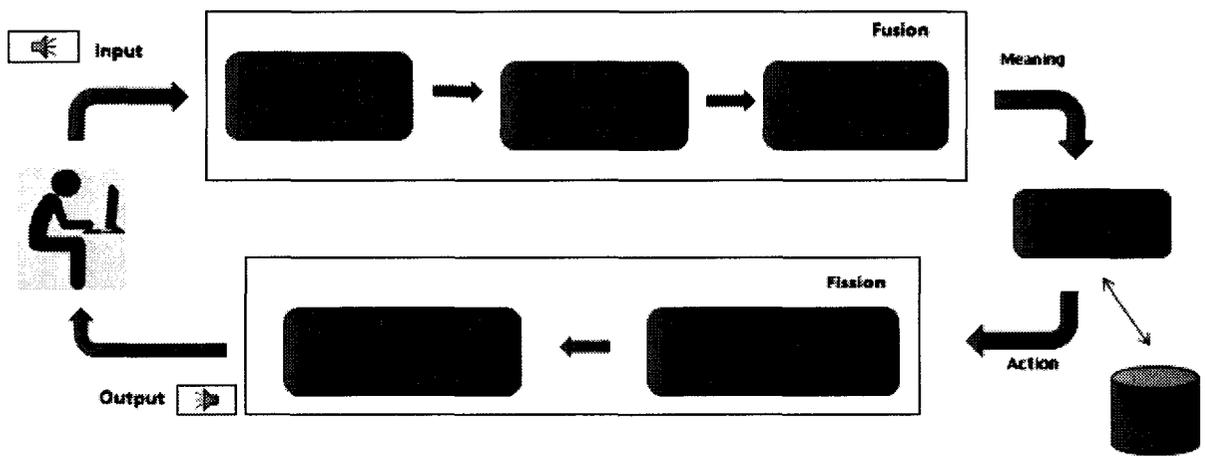


Figure 2.1: General Architecture of Dialogue Management System

A dialogue management system consists of the following components: Input mode, Fusion, Dialogue Manager, Fission and Output mode.

### 2.1.1 Input Mode

Input of a dialogue system can either be speech, text, touch, human gesture etc. Inputs are classified into two categories depending upon the modes. They are *active input mode* such as speech that are deployed by the user intentionally as an explicit command to the computer.

Another, *passive input mode* denotes the behaviour of the user that is recognized by the system which doesn't involve any explicit command to the computer [Bui06].

### **2.1.2 Fusion**

The input from the human/user given to the system is extracted and processed in this level. After it extracts the information from the input, it assigns a semantic representation to the information which can be understandable by the computer. If there are two or more combined input is given by the user, then the information is processed and fused using different levels of fusion. Then, the fused information is integrated and sent to the dialog manager along with the semantic information.

### **2.1.3 Dialogue Manager**

Dialogue manager is considered as the core component of the dialogue management system which is nothing but a program which coordinates the activity of several subcomponents in a system. It acts as the interface to process any task or domain-based actions. The fused information is processed at this level by updating dialogue context. It is also responsible for choosing an action, which results in the change of dialogue state. It also stores and updates dialogue states, decides which dialogue act to perform next and how to continue the dialogue. If any component has insufficient input to perform its task, the dialogue manager gets an alert, which can reconsult the previously invoked component for different input. It also increases the overall performance of the system [Susann98].

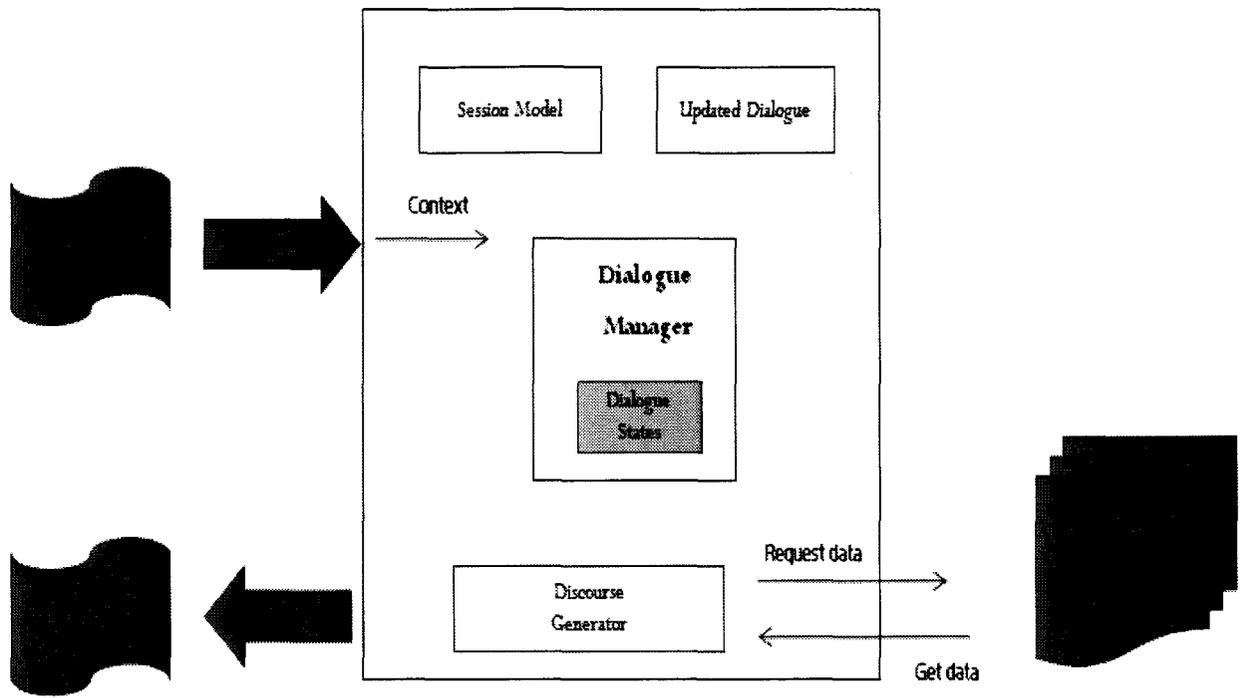


Figure 2.2: General architecture of Dialogue Manager

#### 2.1.4 Fission

Fission is the next component of dialogue systems in which the abstract message is received from the dialogue manager in which the information is in machine understandable format. In fission level, the semantic level information is converted into the output format which is understandable by the user depending upon the input. The fission module is categorized into 3 types [Foster02]: *Content selection and structuring*, where the content is arranged in a structure. *Modality Selection* in which the type of modality is identified according to the user output component. *Output Coordination*, in which different output from the channels are coordinated and sent to the output component of the user.

### **2.1.5 Output Mode**

In this mode, various outputs coordinated in the fission component are displayed to the user depending upon the user's hardware specification. Output modalities include text, speech, gesture, graphics, haptics etc.

### **2.1.6 Multimodal Dialogue management**

Previously, the system processes single user input mode such as text, speech or gesture recognition. Recently, it has been extended to two or more user input modes such as speech and text, speech and gesture, text and gesture, Facial expression, touch, head and body movements etc. in a coordinated manner with multimedia system output [Oviatt02]. Such types of systems are called multimodal dialogue systems.

## **2.2 Review of Existing Approaches**

Dialogue management is classified into different approaches depending upon the task and domain model. According to [McTear02], dialogue management is classified with three types of strategies such as: finite state-based, frame-based and agent-based. In [Xu02], Dialogue management is classified into four categories which is approximately same as the above three categories. In [Cohen97] and [Catizone02], three categories are mentioned namely, dialogue grammars, plan-based approach and cooperative approach which are collectively known as agent-based approaches. Since these approaches are not mutually exclusive and have to combine any two approaches in a dialogue system, researchers come up with a new type of approach which is information state and probabilistic approach. To implement this type of approach, researchers develop a toolkit based on the concept, which can be directly integrated with any type of system.

### 2.2.1 Finite state-based Approach

This approach is the basic and simplest approach used in the dialogue management system. In this approach the system's utterances are represented in the form of state transition network which are pre-determined. According to [Xu02], in this approach both the task model and dialogue model are implicit. In this approach, the dialogue consists of sequence of pre-determining steps. To control the dialogue, system produces prompts at each dialogue state. It can only understand certain words or phrases which are pre-coded in the back end. So, the user has to answer in a single word or phrase in order for the system to produce next action. Since, it contains sequence of steps, until the system complete all the steps it is not possible to verify the previous answer or state.

An example for finite state-based approach is Nuance automatic banking system [McTear02]. The systems is designed with UK based English and it enables the user to conduct bank transactions over the telephone. The dialogue flow is modelled by a set of interconnected dialogue states, in which the system is directed with pre-determined sequence of questions. For example, consider this dialogue flow [McTear02]:

*System: What would you like to do?*

*User: Pay a bill.*

*System: What company would you like to pay?*

*User: Midland Bank.*

*System: How much would you like to pay?*

*User: One hundred and twenty-five pounds fifty-seven pence.*

*System: What date would you like the payment to be made on?*

*User: Monday.*

*System: You want to pay HSBC one hundred fifty-five pounds and fifty seven pence on April 5th? Is this correct?*

*User: Yes.*

*System: Your payment will be sent. Would you like to pay another bill?*

The systems is pre-designed to get three specific value from the user in order to maintain the flow of the transaction, values such as payee, amount and payment date. Sometimes, if the user try to give multiple slot values in one dialogue state, the system fail to update the user's over-informative response and it again prompts a question to answer probably the user has to respond with the same answer once again. For example,

*System: How much would you like to pay?*

*User: One hundred pounds next Monday*

*System: What date would you like the payment to be made on?*

*User: Monday.*

If user wants to complete the transaction very quickly, the system fails to update all the information provided by the user. It understands only pre-set values for a particular question and again proceeds with the sequence of questions. Sometimes, the system fails to correct more than one error at a time, it happens when the user reconfirms multiple values with the system. These types of limitations may lead to a frustrating and unnatural dialogue system. Irrespective of these limitations, Nuance automatic banking system was considered as one of the best applications of Spoken Dialogue System which can handle simple transactions efficiently and effectively.

### **2.2.2 Frame-based Approach**

To overcome the flexibility issue of finite-based approach, frame-based approach is developed as an extension. Instead of continuing the dialogue with some sequence of pre-determined questions, this approach has a form with multiple slots in which the user has to fill in the information. Here, the task model is represented explicitly and the dialogue model is represented implicitly [Bui06].

In [McTear02], an example to frame-based approach, given the Philips automatic train timetable information system. It provides information over the telephone about the trains between German cities. This system is a research prototype and developed in German language. The aim of the dialogue is to enable the system to construct an appropriate database query that retrieves the information required by the user and then to present that information in a suitable form.

[Hulstijn96] developed a theatre booking system and [VanZ96] developed a train timetable enquiry system which relates the entities in the domain to another which has a meaningful structure for the user queries. In [Goddeau96], an E-form (electronic form) type has been discussed for the advertisement which is more complex type of form. The E-form differs from normal type of forms developed in other frame-based systems. These types of systems are basically developed to get one or multiple entries from the user. However, they failed to concentrate more on the transition of dialogue states and more natural dialogue which results the system actions to be very limited and hard to handle complex systems. In [Bohus03], this E-form type is extended to task structure graphs to handle complex transaction over telephone which provides the similar structure used in previous models. This system is designed to determine the behaviour of dialog control and language understanding module. The applications of this approach include various toolkits such as RAD, UNISYS's Dialogue Design Assistant (DDA), etc.

### **2.2.3 Information state-based Approach**

To overcome the limitations of previous approaches, Information state-based approach is a dialogue theory with five different components; each has its own functionality [Traum03]. *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 [Larsson20] and EDIS [LarsTrau20]. Several other applications of this approach include MATCH system for multimodal city help [Johnston02], Virtual Music Center [Hofs03], etc.

#### **2.2.4 Probabilistic Approach**

This approach is an extension of Information state approach which uses probabilistic techniques. The techniques include Markov Decision Process (MDP) or Partially Observable Markov Decision Process (POMDP). The basic idea is to overcome the limitations of Multi-layer 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 modelled 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 [Singh02]. In [Young99], dynamic programming, Q-learning or sampling-based reinforcement learning is used to optimize the dialogue cost function. In [Leco01], 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.

### **2.2.5 Plan-based Approach**

This approach is developed to overcome the complexity of dialogue modelling in the previous approaches. In this approach, the main task or goal of the user is discovered and respond appropriately to the user's requirement. The dialogue acts here are considered as speech acts. The basic idea of this approach is that, the main job of the system is to identify and respond to the user plan and the user's speech acts are considered as part of the plan. In Verbmobil project, the dialogue is divided in sub-goals by a set of plan operators derived for the example dialogues [Churcher97]. This approach is criticized on practical and theoretical experiments. In some of the cases, it is even hard to predict how this system understands the actual goal of the user. Sometimes, it lacks to handle plans or goals of the dialogue model. In some cases, there is no actual specification that what the system should do, that is the actual interpretation is quiet different form the illocutionary acts.

To overcome these types of issues, a conversational games theory is developed. This extension combines the dialogue grammars and plan-based approaches by including a goal in its structure, in order to provide more natural language dialogue. It actually handles the actual human-computer dialogue in a task-oriented basis. A task-oriented dialogue consists of one or more levels of conversations, each represents a task. Since, this extension of approach mainly developed for games, it allows two or more games to be embedded on one game. It consists of an opening move and an ending move with double checking options and side sequences. Applications developed on the basis of this approach includes, SUNDIAL (Speech UNderstanding in DIALogue), TRAINS-96, etc.

### **2.2.6 Agent-based Approach**

This approach is developed on the basis of viewing dialogue management as collaborative process between intelligent agents. In this approach, both the user and the system are the agents and the interaction and dialogue process takes place by mutual understanding between two agents in a simulated environment. Here, it is designed to capture the motivations behind the dialogue mechanisms and the discourse phenomena such as confirmation and clarification. Here the intentions or goal of the agents are modelled as beliefs and the main goal of this type of system is that the goal of the agent is shared as beliefs among the agents and the number of agents involved in this system work cooperatively to achieve the appropriate goal. It also uses several techniques used by the plan-based approach and previous approaches. Though this approach can handle complex applications, it uses the concept of dialogue grammars and combined techniques of plan-based approach to understand the goal of the agent in the environment. So, it cannot handle more complex applications effectively and efficiently. The application of this approach includes COLLAGEN, TRIPS, ViewGen, etc.

### **2.3 Contextual Control Model**

In the Human-Computer Interaction, we determine the human performance such as actions, goals etc. by the context or the situation of the dialogue. Sometimes, to choose a particular action for the user response, the system should not only understand the intention of the user but also the current situation of the dialogue between the user and the system. Even there are lot of techniques been developed to determine the user goals and actions, but to choose action according to the situation in a dynamic environment, Hollnagel came out with new idea of Contextual Control Model in 1993. These types of systems work on the basis of assumption

for every situation to make an action. This model is developed based on three concepts namely, *competence*, *control*, and *constructs* [COCOM].

- Competence represents the possible actions the system can take to respond to the user depending upon the situation according to the user requirements.
- Control represents the characteristics and the performance of the competence and the way it is applied. This model deliberately sets four control modes namely scrambled, opportunistic, tactical and strategic. These modes changes from one mode to another depend on the situation in an application. In some situation, these modes range from no control at all to completely deterministic performance. Another mode has to do characteristic performance in a particular mode, that is how the actions are determined.
- Constructs represents what and how the system assumes about the particular situation in which the actions take place. This feature is included in this model for selecting actions and interpreting information.

The important part of this model is planning and the goal should be achieved in given time-horizon. The planning is influenced by the context and knowledge between the actions or expectations about the situation. The sequences of actions are the outcome of the planning which is constructed rather than pre-defined.

### **2.3.1 Four Control Modes**

The control modes are modelled depending upon their characteristics and performance in the model. Each control mode is associated with its characteristics and type of performance. These modes of team behavior vary in terms of the degree of forward planning. The following are the four control modes [COCOM].

- ***Scrambled mode:*** In this mode, the choice of next action is basically irrational or completely unpredictable. The type of performance is thus, paradoxically characterized by the lack or absence of any control. In this case, the situation is paralysed and zero in control, so no corresponding actions will be taken which means the situation is out of control from planning.
- ***Opportunistic mode:*** In this mode, the next action is predictable depending upon the current context. Here the information is inadequate and due to less competence, planning and time is limited and the context is not completely understood. In that situation, there is a possibility of making useless attempts because of the choice of choosing actions are limited. The application will be in unusual state of environment. It will not be able to fully assess the situation, often having difficulty finding and assessing relevant aspects of the environment. For example: providing 'one-click' mechanisms to delay or cancel it.
- ***Tactical mode:*** In this mode, the performance of the application follows a certain procedure corresponds to the situation. The amount of information sought in this mode is expected to be beyond what is immediately observable, but may be limited to what routine procedure requires. Here, the planning is also limited and sometimes the more features are taken into account in order to select appropriate action for the situation. For example: Double checking the solution and provide feedback to the agent on a set of evaluation criteria.
- ***Strategic mode:*** This mode has the higher level of control and concentrates on the long term planning (global view). The amount of information sought and coordination required between the user and the machine is expected to be extensive. It compares

multiple feasible solutions and iterates several times to make the “best” decision possible. It has less influence on choosing choice of actions. For example, in airline support system, the system may compute “optimal” solutions to minimize passenger delay, maximize aircraft usage, etc.

The Contextual Control Model is used in developing and testing various applications such as Airline rescheduling tasks, for dynamic decision making in Airline operations to improve airline recovery from irregular operations, assessing team behaviour in a human supervisory control task, human reliability analysis, in traffic environment to track the single driver behaviour, etc.

## **2.4 Conclusion**

Various approaches have been developed to make the dialogue systems to provide services efficiently and effectively. Also, the complexity increases with the increase in number of approaches, dialogue model and domain models. We have mentioned various approaches from finite state-based to agent based which has the simple structure and less complexity. But all such systems fail to handle critical situations and completely understand the actual intention of the user because of its construction and pre-defined response. This chapter address the approaches to achieve goal and select actions according to the user requirement. Also, Contextual Control model also discussed for planning and select actions according to the situation and user requirements.

## Chapter 3

### Literature Review of Dialogue Management Techniques

Dialogue manager is core component of the dialogue management system as it is responsible for analysing user goals and decision making. This chapter gives a brief discussion about various techniques used in the dialogue manager to understand user's intention to provide required services. Since the decision making mechanism in our model is dominated by this contextual control model, we have discussed some of its previous works at the end of this chapter.

#### 3.1 Dialogue Management Techniques

##### 3.1.1 Handcrafted Dialogue Managers

When the researchers started concentrating on spoken dialogue systems, they designed a handcraft dialogue manager to map the system states to the system actions and maintain one dialogue state. Handcrafting process was very time-consuming and more effective in care of handling speech act errors. Historically, the issue of uncertainty arises in dialogue systems, the main goal of the researchers were to make the system interact with the human effectively. Various applications are developed using this handcrafted design and developed using high-level specification language such as VoiceXML [Balantine99]. This type of dialogue managers are specified as a policy graph which is a finite-state controller consisting of set of nodes  $M$ . Each controller node is assigned an action  $M \rightarrow A$ . Arcs are labelled with an observation and each controller node has an observation represented by an outgoing arc.

[Jason05] created three handcrafted policies for dialogue managers known as HC1, HC2 and HC3. A logical diagram of HC1 handcrafted dialogue controller is shown below.

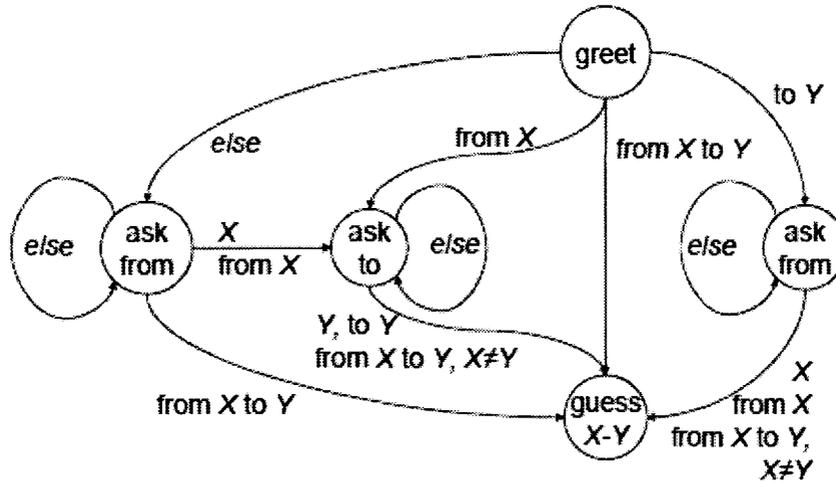


Figure 3.1: HC1 handcrafted Dialogue Controller

HC2 and HC3 handcrafted policies are the extension of HC1 to overcome its limitations. The system start with the action *greet* in all three handcrafted controllers. The actions are performed to fill the *from* and *to* fields without any confirmation. It repeats the same action again and again till it receives a response from the user which is not sensible. HC2 handcrafted controllers overcome this issue by taking a fail action immediately if it receives any non-sensible observation. Other than this feature, HC2 is identical to HC1. HC3 is also same as HC1 and HC2, except it confirms each response from the user while filling the slots. The policy graph uses value function to represent the return from the explicit node. But, in this case the policy graph does not make an expected return from the nodes. [Hasen98] solve a linear equation to solve the expected return from each controller node given in V,

$$V_m(s) = \text{Immediate reward} + \text{discount} * \text{future reward}$$

$$V_m(s) = r(s, \pi_{FSC}(m)) + \gamma \sum_{s'} \sum_{o'} P(s' | s, \pi_{FSC}(m)) P(o' | s', \pi_{FSC}(m)) V_{l(m,o')}(s')$$

The expected value for starting node  $m$  and belief state  $b$  can be calculated by evaluating  $\sum_s V_m(s)b(s)$ . The result of the above equation leads a set of vectors; one vector for each node gives the expected return. The belief state  $b$  can be calculated from  $m^*$ ,

$$m^* = \operatorname{argmax}_m \sum_s V_m(s)b(s)$$

Even though this dialogue manager is handcrafted and simple in its construction and handling, it has lot of limitations to overcome such as maintaining one dialogue state leads to an issue in handling complex applications. In spoken dialogue system, it performs least well as it fails to handle speech act errors and uncertainty in the application.

### **3.1.2 Bayesian Network for Dialogue Management**

A Bayesian Network (*BN*) is a graphical model used to describe dependencies in a multivariate probability distribution function (pdf) defined over a set of random variables [Plamen03]. Since researcher couldn't handle complexity of developing dialogue systems, they tried to use Bayesian network in dialogue management which is also a handcrafted technique in order to have a more sophisticated human- computer interaction. In 2003, [Plamen03] proposed Bayesian Networks method to develop a spoken dialogue system for tour-guide robot and visitors in mass exhibition condition. An interpretation of user goal at each dialogue state is clearly determined in this application. Bayesian network is introduced to infer the exact goal of the user which exhibit to attend next. Since, it is difficult to predict the user's intention in an exhibit as more crowds will be gathered and more chance of speech recognition errors considering acoustic conditions. A laser scanner has been used to detect the unaffected data assuming user's intentions as dependent data patterns. As such other spoken dialogue systems, the tour-guide takes the initiative to start the dialogue process and observes the user's behaviour which is modelled as user goals in this application. Bayesian

network is used in the form of directed acyclic graph which consists of nodes and arcs. Nodes correspond to the variable and arcs represent the conditional dependency assumption between the variables. Plamen further calculated a continuous probability distribution from the data for all parent nodes in order to map the exact interference on each node. They also defined few variables to calculate the probability distribution over the dialogue states. In [Prodanov04], error handling for this application is introduced as an extension to the prior work. He stated that during interaction between the visitor and the tour guide, the speech utterances can be mapped into dialogue states depending upon the user goals which are used to infer next dialogue state. He has also given a graphical representation of the process in which the *UG* stands for the user goal and *DS* for the dialogue state. Initially, the dialogue state is at time  $t$  and user goal at time  $t-1$  which can also affect the user goal at time  $t$ .

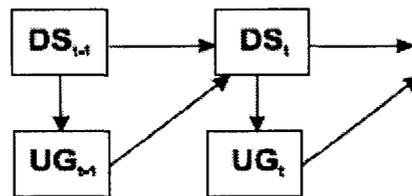


Figure 3.2: Dependency graph for Spoken Dialogue Management

Here, tour guide has to interpret the visitor behaviour to user goals which corresponds to each dialogue state in low-level behavioural events. The basic connection of Bayesian Network in this application is provided by the evidence that the initiated variable can pass through the series of diverging connections until an intermediate variable is available. If there exists, a common child for two or more parents nodes which can affect the state of the other parent node. [Sang09] developed a Skilligent robot to control the robot behaviour during run-time. Sub-goals are determined first by clustering similar features of state transition tuples which are composed of current state, action and next state. They developed a method for

reconstructing learned networks and increment learning for increasing sub goals. They further developed a simulated environment with skills like Dribbling-Box-Into-a-Goal (DBIG) and Obstacle-Avoidance- While-Dribbling-Box (OAWDB) to validate the methods. [Prodanov05] proposed a probabilistic model based on a Bayesian network framework for error handling in human-robot spoken dialogue systems under adverse audio conditions. As the system is based on different multimodal information sources with a complex combination of spoken dialogue system, it fails to process the sub goals which in some exceptional case, the entire system has to be reconstructed.

### **3.1.3 Supervised Learning and Markov Decision Process**

Considering more complex systems, handcrafted dialogue managers needs iterative process to test and is more expensive. Then, the action taken by the system may also affect the long – time goal of the system for some unusual situation. In order to choose the actions  $A_m$  automatically, researcher came out with the idea of Supervised learning and Markov-Decision process.

*Supervised learning* is a simplification techniques in which we can maintain the single dialogue and the actions are learnt from the corpus. Though this techniques has lot of limitations due to maintain single dialogue state, creating a dialog policy is difficult because of collecting training data for the corpus. It simply learns the approximate data from a corpus to create a policy, so the overall performance of the system will be limited. If wizard style dialogues are included in the system, selecting proper actions at each time step is a tedious process using this technique. It always tries to map the machine state  $S_m$  directly to the action  $A_m$ .

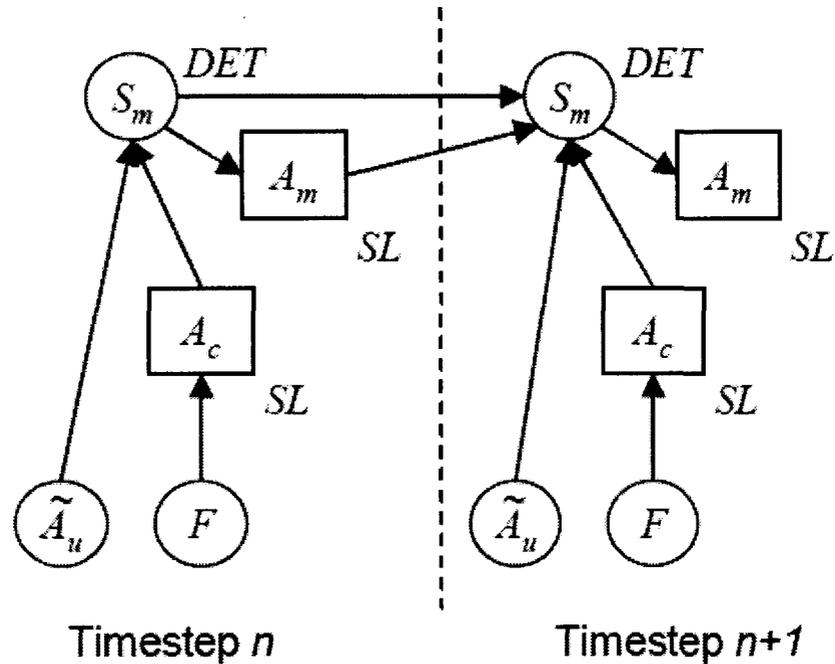


Figure 3.3: Supervised Learning for action selection

In the figure, the node  $A_m$  is trained on a corpus of dialogue states. A decision node  $A_c$  is introduced to state which action to be taken next. Though supervised learning techniques are a simple one, it does not perform effectively in human computer dialogue as the actions are learnt from a corpus of training data. It will be quite expensive to test and maintain applications developed using this technique.

Due to increase in complexity and level of uncertainty in decision making process, there arise many questions which are not answerable. How to choose best actions which won't affect the long- term goal? Sometimes, the current action gives a solution for long-term goal but affects fewer states. How to calculate the exact value where the actions are affected? How to make an assumption that this action doesn't affect the immediate and long-term goals? How to solve the decision problem? Markov decision process takes care of all this issues and automates all the processes. E. Levin et. al, introduced the *Markov Decision Process* for dialogue management for ATIS Air travel domain in 1997 [Levin97]. MDP

consists of four components namely, states, actions, transitions and rewards. The transition in MDP takes care of which action to take for each state. Also, we have to mention the effective of action for each state. It maintains a probability distribution over the dialogue states. Rewards are the measures to compare different actions. The rewards are calculated for each system action for each dialogue state. The solution of mdp is policy ( $\pi$ ) which represents the best action taken for each dialogue state for which the next resulting state, for each state and action are specified.

Let  $S$  be the set the states and for each state  $s \in S$ , there is an action taken  $A_s$ . Since, the current state is known and the policy is represented as function of states which is given by,  $\pi: S \rightarrow A$ . At time  $t$ , the system is initiated to enter a state  $s_t$  and choose an action  $a_t$  determined by the policy  $\pi$ . For each action at time  $t$ , a reward is assigned to  $\pi$ . If the system choose the best action, it will receive a positive reward else negative reward to choosing wrong actions for each state. The system transitions from one state to another at time  $t+1$  from  $t$  and so on. MDP maintains a probability over this transitions which is given by,

$$T(s', a, s) = P(s_{t+1} = s' | s_t = s, a_t = a)$$

Where,  $T(s', a, s)$  denotes the probability to enter state  $s'$  at time  $t + 1$ , given state  $s$  and action  $a$  at time  $t$ . For deterministic transitions and actions corresponding probabilities can take only 0 or 1 values. The actual goal of the system is to find the policy  $\pi'$ , which maximizes the total reward [Ragni06].

$$R = \sum_{r=0}^{T-1} r_{r+1}$$

In [Levin20], MDP approach has been tested for learning the strategy of airline task information system (ATIS). They used the concept of ‘Day and Month Dialog’, in which the system has to get the day and month values exactly from the user in shortest time and in few

interactions to reserve a ticket. To accomplish this, the objective function is introduced as follows.

$$C = W_i \langle N_i \rangle + W_r \langle N_r \rangle + W_o \langle f_o(N_o) \rangle + W_s \langle F_s \rangle$$

Where, C is the expected cost to achieve the goal and W is the weight to determine the tradeoffs between the costs.  $N_i$  is the number of interactions,  $N_r$  denotes the number of errors obtained from expected value and  $N_f$  is the duration of achieving the goal. A set of possible actions are also given to the system for the day and month values. While executing the system, it first executes the actions for the values and it then it starts the speech recognition system to get the user response. The description of the system is as follows:

```

Initialization:  $S_{t=0} = S_I$ 

For Each Iteration  $t$ : {
    if ( $S_t \neq S_F$ ) {
        compute current action  $a_t$  according to the strategy
        execute  $a_t$ 
        update current state
         $t = t + 1$ 
    }
    else
        END
}

```

Figure 3.4: Sequential decision Process of ATIS

A dialogue strategy of MDP specifies what action to be performed next for each state. Here, they introduced two assumptions to describe the dialogue system as MDP to find the

optimal strategy. The first assumption is that assigning probabilistic model to the state transitions. According to transition probabilities, while in state  $s_t$  at time  $t$  an action is taken  $a_t$ . Then it transitions to  $s_{t+1}$  state which is given by,

$$P(s_{t+1} | s_t, s_{t-1}, \dots, s_0, a_t, a_{t-1}, \dots, a_0) = P(s_{t+1} | s_t, a_t)$$

The second assumption relates to the modelling cost of the system in which the system receives a feedback cost  $c_t$  when an action  $a_t$  is executed at state  $s_t$ .

$$P(c_t | s_t, s_{t-1}, \dots, s_0, a_t, a_{t-1}, \dots, a_0) = P(c_t | s_t, a_t)$$

The dialogue session mentioned in this model is the path in the state space between the initial and end state which is illustrated in the following diagram [Levin20].

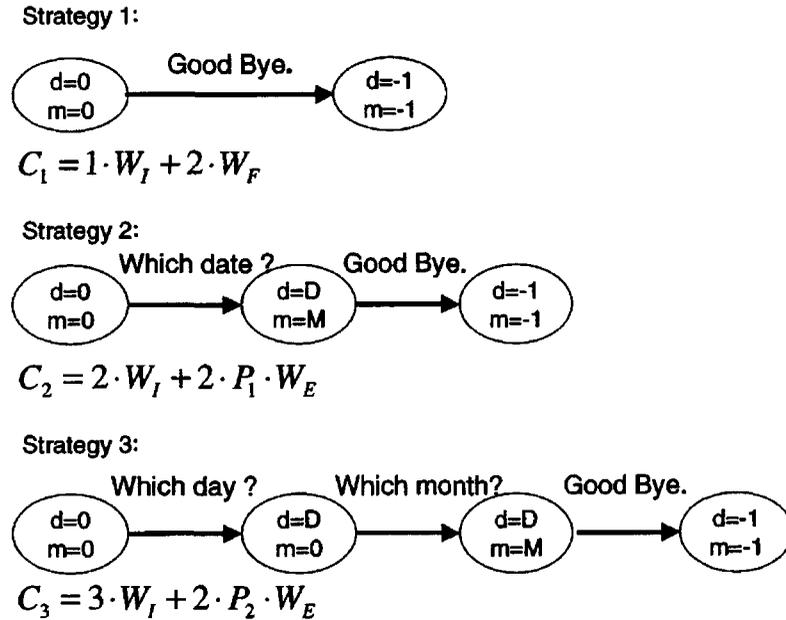


Figure 3.5: Three possible strategies of ATIS Spoken Dialog System.

[Young20] introduced a probabilities framework for spoken dialogue systems by making an assumption on the behaviour of the system as Markov Decision Process. He used sampling method to obtain optimal strategy for the model. [Roy20] proposed a method to handle noisy and ambiguous utterances from the user by inverting the dialogue state to represent the user intentions. He used a mobile robot called Florence Nightingale (FLO), a nursing home assistant to carry out experiments. The MDP is specified by the following,

- A set of states  $S_E \{s_1, s_2, \dots, s_m\}$
- A set of actions  $A_E \{a_1, a_2, \dots, a_m\}$
- A set of transition probabilities  $T(s', a, s) = P(s' | s, a)$
- A set of rewards  $R : S \times A \rightarrow R$
- An initial state  $s_0$

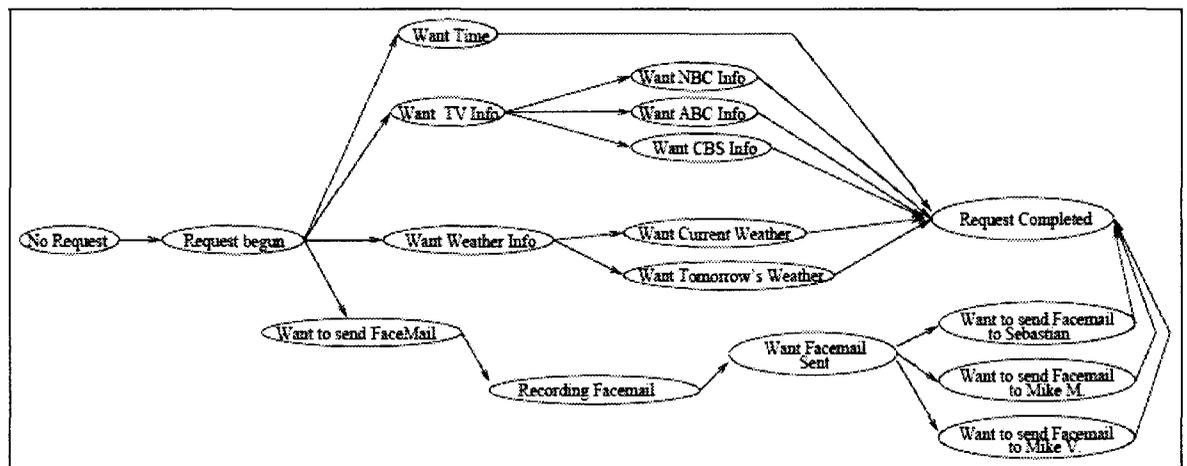


Figure 3.6: A graph of basic MDP underlying the Dialog Manager

With the increase in different methods of MDP, increases the complexity and limitations. Even though, MDP try to find the optimal strategy of the application by interacting effectively with the users, there are fond of reacting to recognition errors and lack in domain knowledge. The main drawback of MDP is that, it makes decisions based on the current state alone which cannot be possible in real-world applications.

#### **3.1.4 Semi-Markov Decision Process**

[Cuayahuitl09] proposed a hierarchical reinforcement learning to evaluate the dialogue behaviours of a simulated agent in a simulated environment. He used Semi-Markov Decision Process to learn the dialogue behaviours in a realistic conversational environment. He mainly concentrated on the dialog manager part of the spoken dialogue system as the main principle of the system is to choose best action which results in the change of dialogue states. The agent in this model learns the behaviour by interacting with the environment, maximizes the reward model, where the actions are mapped to situations by aiming long-term goal. They solve an MDP by mapping the current state  $s_t$  to an action  $a_t$  correspond to the dialogue policy  $\pi^*(s_t) = \arg \max Q^*(s_t, a_t)$ . The Q function denoted the cumulative rewards for each state – action pair. His proposed method generated both coherent and distorted conversations which are used as test bed for information-seeking spoken dialogue systems. It consists of two simulation models namely, ASR error simulation and Simulated user behaviour. The figure 9 shows the execution of simulated environment. It tracks the machine behaviour using the dialogue strategy and chooses actions accordingly, updating the conversational history in the knowledge base.

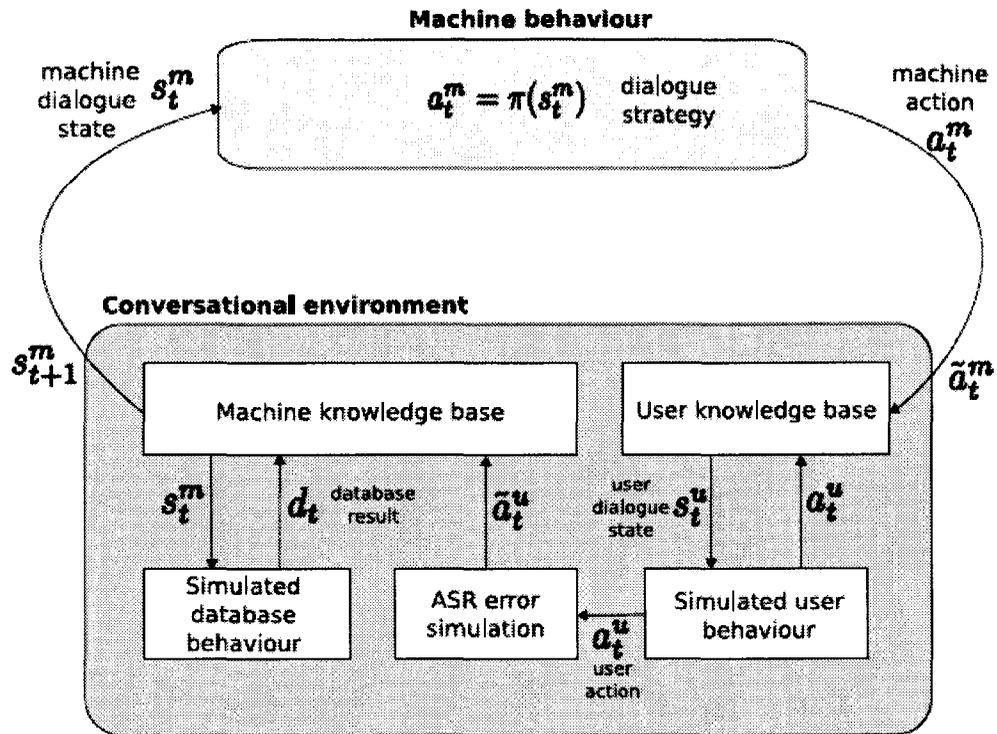


Figure 3.7: The simulated environment interaction for Human-Machine conversations.

A human-machine dialogue is modelled by the perception and actions of both conversant. The user respond with errors are distorted into ASR error simulation and then it is sent to the knowledge base. The conversant at time  $t$ , observes the current knowledge state, selects the appropriate dialogue type to respond, update the knowledge rich state. The dialogue control is modelled as Semi-Markov Decision Process (SMDP) in discrete time is denoted as  $M = (S, A, T, R)$ .  $S$  denotes the set of states,  $A$  denotes the set of actions,  $T$  denotes the transition from state  $s$  to next state  $s'$  with a probability  $P(s', \tau | s, a)$  and  $R$  denotes the reward function for choosing an action. The variable  $\tau$  denotes the time-steps taken to execute an action  $a$  in a state  $s$ .

This method is applied to flight booking domain and travel planning domain to carry out several levels of testing to learn the dialogue agents outperform in deterministic behaviour in realistic environment. Even though there is lack in training data, finally they proved that simulated agents make fewer errors in realistic environment.

### **3.1.5 Partially Observable Markov Decision Process**

Partially Observable Markov Decision Process (POMDP) is an extension to Markov Decision Process, which is introduced into dialogue management to overcome the issues addressed by MDPs such as uncertainty, choosing actions upon current state and experiencing speech act errors in noisy environment. Pomdp concept was developed in 1965 by Astrom, which was basically used in the field of engineering. Later, researchers incorporate this concept to Artificial Intelligence for analysing the behaviour of agents or robot using sensors. Since MDP failed to handle ambiguous data, researchers tried to incorporate Pomdp concept into dialogue management as it maintains probability distribution over MDP dialogue states called belief state and set of observations to choose a best action  $a_m$  for each dialogue state  $s_t$  at particular time  $t$ . The dialog policy  $\pi$  is based on the belief state rather than the underlying state. [Roy20] used a nursing home assistant robot application to compare the results of MDP and POMDP and proved that POMDP performs better than MDP in some situations and in noisy environment. The following figure shows the different between conventional and probabilistic dialog managers [Young06].

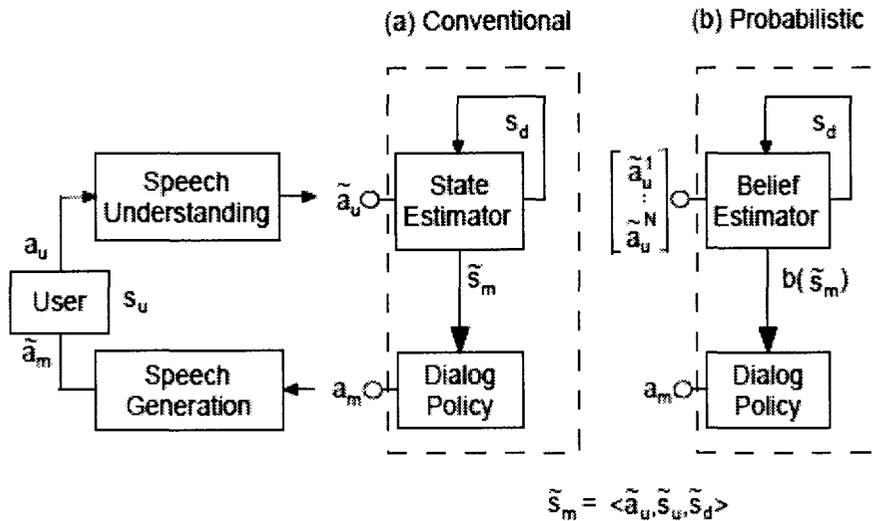


Figure 3.8: Structure of a Spoken Dialog System

The above figure shows that the conventional dialogue manager maintains a single state estimation, whereas the probabilistic dialogue manager maintains a distribution over all dialogue states. The tilde on the system state and actions represents some noise in environment. This framework provides a mechanism for modelling uncertainty, i.e., what actually the user requires.

POMDP is defines the set of tuples same as MDP, in addition it has observation O and observation function Z which is given as  $\{S, A, T, R, O, Z\}$ .

S – set of states

$A_m$  - set of actions that a machine may take

T - transition probability,  $T(s' | s, a)$

R - reward for each action,

O - set of observations

Z - observation probability,  $p(o | s', a)$

$b(s)$  – probability of being in one state.

$\lambda$  – Discount factor ( $0 \leq \lambda \leq 1$ )

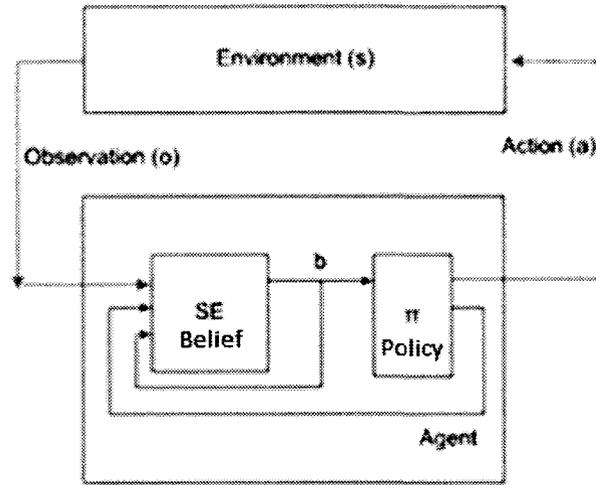


Figure 3.9: The agent – Environment Interaction

At time  $t$ , the systems will be in some unobserved state  $s \in S$  [Young06]. The initial state  $s$  is not known exactly, so Pomdp based systems maintains a probability distribution over the machine states at time  $t$  is known as belief state,  $b(s)$ . The system selects an action  $a \in A$ , based on the current belief state  $b$  and receives a reward  $r(s, a)$  and transitions to a new unobserved state  $s'$ . The system receives an observation  $o' \in O$  depends on  $s'$  and  $a$ . Finally, the belief state  $b$  is updated based on  $o'$  and  $a$  at a time step  $t$ , which is illustrated below.

$$b'(s') = p(s'|o', a, b)$$

$$= \frac{p(o'|s', a, b)p(s'|a, b)}{p(o'|a, b)}$$

$$\begin{aligned}
&= \frac{p(o'|s', a) \sum_{s \in S} p(s'|a, b, s) p(s|a, b)}{p(o'|a, b)} \\
&= \frac{p(o'|s', a) \sum_{s \in S} p(s'|a, s) b(s)}{p(o'|a, b)}.
\end{aligned}$$

The machine receives a reward  $r(b_t, a_{m_t})$  at each time step  $t$ . So, the expected reward over belief states is given by,

$$\rho(b_t, a_{m_t}) = \sum_{s \in S} b_t(s) r(s, a_{m_t})$$

The return  $R$  is computed by cumulative, infinite horizon and discounted reward.

$$R = \sum_{t=0}^{\infty} \lambda^t \rho(b_t, a_{m_t}) = \sum_{t=0}^{\infty} \lambda^t \sum_{s \in S} b_t(s) r(s, a_{m_t})$$

[Roy20] proposed a probabilistic technique for spoken dialogue management. He compared MDP and POMDP by applying on a home nursing assistant robot. First, he applied MDP method to analyse how the system performs in the noisy environment. Moreover, the voice reliability of the mobile robot is poor; MDP does not accurately understand the user goals and the responses are not related to the user requirement. The solution of MDP is neither sufficient nor robust. To increase the performance of the mobile robot and perform partial observability on the dialogue states, Roy et. al introduced POMDP to improve the performance of mobile robot. He also track step by step process of performance of POMDP reacting to the dialog states. He also tested with noisy environment, but the results of belief state values and confidence scores remain the same. Further, it replaces the initial state  $s_o$  with initial belief state,  $P(s_o : s_o \in S)$ . During planning phase it finds an optimal strategy for

all belief states. The performance is reduced when dealing with uncertainty and finding optimal policy for all POMDP applications are computationally expensive. He further introduced Augmented POMDP by applying some restriction on the belief states in order to deal with uncertainty problems.

[Pineau03] introduces the point-based value iteration algorithms (PBVI) for POMDP planning. He selected few belief values, evaluated an exact value iteration solution and tracked the solution for only those selected belief points. He applied this technique in a Laser tag application and the results showed that PBVI for POMDP can be applied for large systems.

In 2004, Poupart introduced a scalable algorithm for large POMDPs by combining Value Directed Compression (VDC) technique with Bounded Policy Iteration (BPI). He added that value functions of very good policies are often represented using small vectors. So, the application can perform better only if the dialogue states are less than 1000. By using this technique for POMDP, the scalability issue and complexity of the policies are reduced. He tested these techniques with some network management problem which has 33 million dialogue states [Poupart04].

In 2005, (Jason, Poupart and Young) introduced a framework for discrete and continuous observation component for POMDP. They used a simulated dialogue test bed for travel domain to prove that the optimal policy for continuous POMDP outperforms previous traditional approaches. Also, the proposed method improved the performance of handcrafted dialogue manager by incorporating confidence score and belief monitoring. This method takes additional information of confidence score into account and creates a policy for continuous POMDP solution, whereas the discrete POMDP doesn't use confidence score

information for planning. By contrast, MDP uses lot of confidence score information for planning but does not have belief monitoring [William05]. The influence diagram of continuous POMDP is given below.

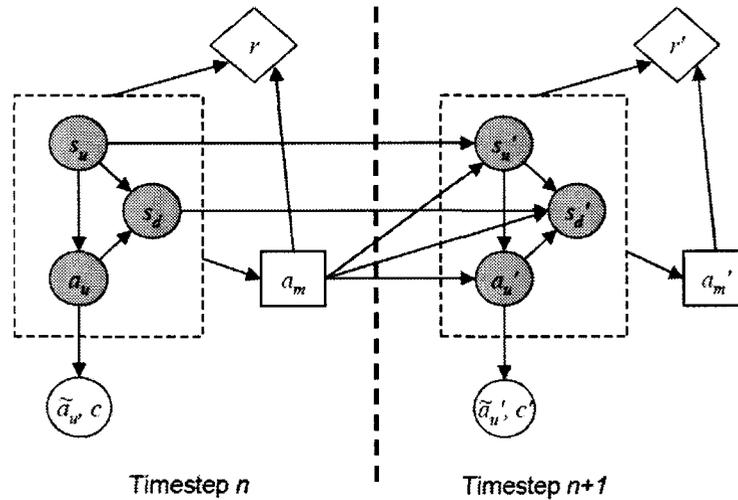


Figure 3.10: Influence diagram of continuous POMDP

In the above figure, the shaded circle shows that the system is in unobserved state and the dotted box indicates the composite POMDP state,  $s$ . The tilde symbol in user action  $a_u$  denotes that the user may in noisy environment. The system actions  $a_m$  depends on the belief state  $b(s)$ , not the unobserved state.

In 2006, William et. al. extended the POMDP with continuous observation for Pizza ordering domain. The point-based value iteration algorithm used to measure the performance of this system is Perseus [Spaan04]. To speed up the system processing, Perseus heuristically selects a small set of representative belief points. They evaluated the confidence score by assuming two distributions namely, correct and incorrect recognition. Figure 13, shows an example conversation between user and the system for Pizza-ordering domain. The first

column shows a sample dialogue, confidence score and how the speech recognisor understand the utterance (shown in brackets). The second column shows how the dialogue manager understands the user utterance by tracking the dialogue states and confidence bucket. The third column shows how the POMDP belief state tracks the same dialogue [William06].

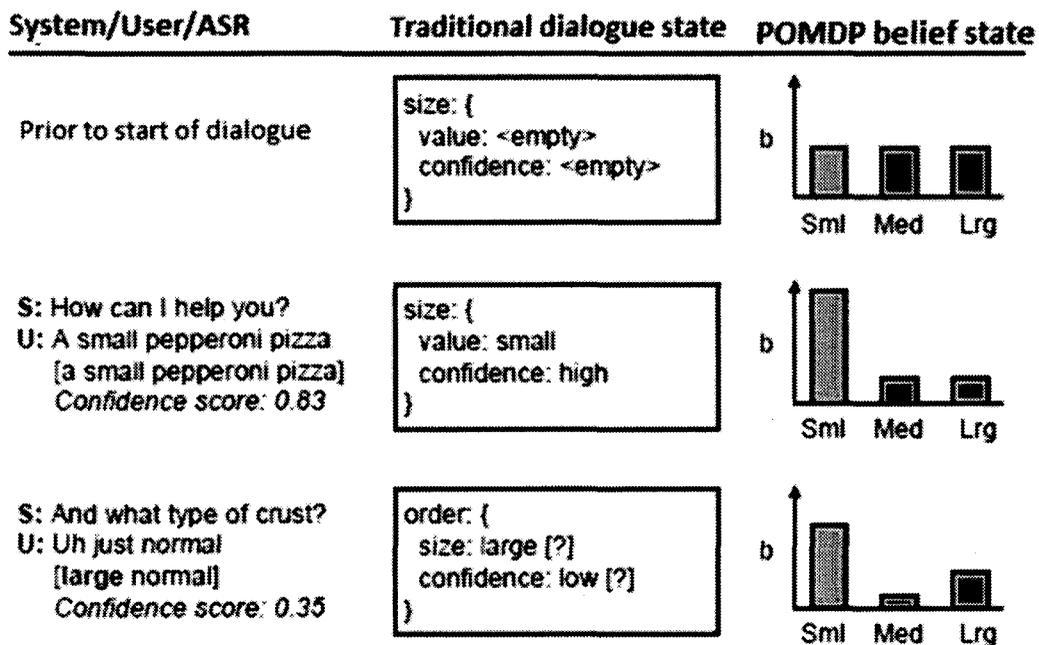


Figure 3.11: Example conversation in Pizza- Ordering domain.

In 2007, Bui with his colleagues proposed a DDN-POMDP approach which focuses on real- time online belief state for affective frame-based models. Their experiments proved that, this approach outperforms 3 handcrafted dialogue managers, handle large number of slots and able to keep track of user’s affective state [Bui07]. [Thomson07] proposed a new framework for training real world POMDP based dialogue system. His approach discuss

about the new policy optimization based on grid- based Q-learning with a summary of belief space. This type of system allows real time conversation between the user and the system by recording new dialogues. In summary POMDP, the original actions and belief space are called master actions and master space respectively, while the summary space actions and belief space are called summary actions and summary space respectively. When the conversation starts between the user and system, the system chooses an action at each point of time which is mapped to summary belief space. The nearest summary point in the grid is determined and the optimal summary action given by that point is chosen. The experiments showed that this policy works well with human user as it performed 90% well even without accessing the training data.

Bui developed a practical dialogue manager using POMDP in 2007 [Bui et. al.07]. They applied factored POMDP model in three applications such as QA dialogue system, Virtual guide and route navigation system. The basic goal is to improve the handcrafted policies with POMDP approach. To accomplish the task, they developed a software toolkit to conduct experiments and track the performance of this approach. The results of these applications are compared with the handcrafted dialog managers using two different solvers to evaluate the performance of the system. In 2008, Jason extended the POMDP approach for tracking multiple dialogue states using ASR N-best list [Jason08]. In 2010, Thomson and Young proposed a new framework of POMDP known as Bayesian update of dialogue state using loopy belief propagation (LBP) algorithm. This approach has been the most effective of spoken dialog systems because the prompts can be designed to elicit highly restricted user responses. However, the choice questions for the user are limited. This approach has been tested with both simulated and real user environment.

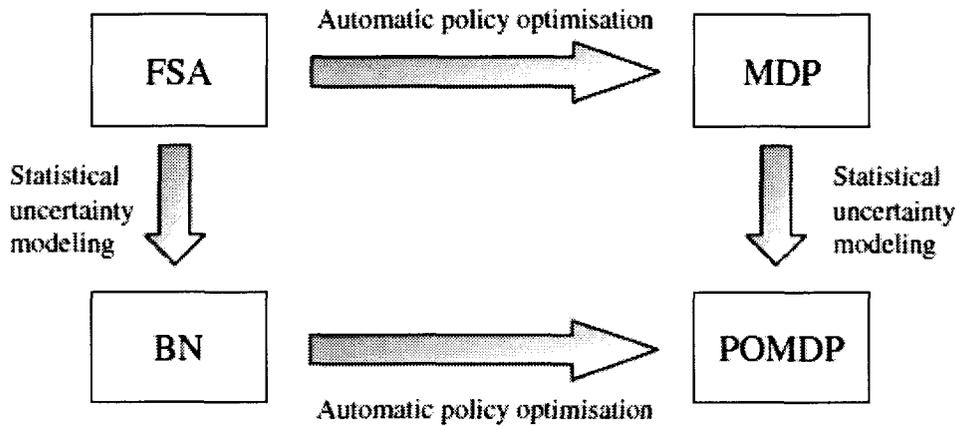


Figure 3.12: Frameworks for modeling uncertainty and policy optimisation in spoken dialogue systems.

This method improves the standard algorithms and made belief space update tractable. It also uses Natural Actor Critic (NAC) algorithm of [Peters et al. 05] for optimising policies of POMDP dialogue manager over a factorised state space.

### 3.1.5.1 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  $s \in S$  into three components such as: 1) the user's goal,  $s_u \in S_u$ ; 2) the user's action  $a_u \in A_u$ ; 3) history / state of the dialogue  $s_d \in S_d$ . 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 [Jason05].

- 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  $a_u$ , 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 penalise asking an open question.

The POMDP action  $a_m \in A_m$  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  $a_u$ . The factored POMDP is given by decomposing the POMDP transition function which is as follows:

$$\begin{aligned}
 p(s' | s, a_m) &= p(s'_u, s'_d, a'_u | s_u, s_d, a_u, a_m) \\
 &= p(s'_u | s_u, s_d, a_u, a_m) p(a'_u | s'_u, s_u, s_d, a_u, a_m) p(s'_d | a'_u, s'_u, s_u, s_d, a_u, a_m)
 \end{aligned}$$

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.

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

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.

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

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.

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

Thus, the transition function of POMDP is given by,

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

The observation function of POMDP is given by,

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

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.

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

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 [Young06]. 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.

### **3.1.5.2 Hidden Information State model**

In 2005, Young et al. proposed a model to integrate the knowledge representations with the appropriate statistical model known as Hidden Information State (HIS) model for statistical dialogue systems. The main idea behind this model is that a belief state distribution can be represented efficiently by partitioning the states and grouping them together for extremely large state space [Young et al.05]. The HIS model deals with two components of factored POMDP; User goal and user action components. In user action model, the value of the previous user action is not required to apply the update belief state equation if the system is memoryless. The user goal model targets on the database inquiry applications in which the user goal is deemed to a specific entity. User goal partitions are represented as forest of trees where each tree represents a single partition which has been illustrated in the following figure 15. Each user utterance is decoded into an N-best list of dialog acts and each incoming act plus the previous system act are matched against the forest of user goals and partitions are split as needed [Young06]. The space of all user goals are described as a set of ontology rules and these rules specifies the hierarchical structure of data. When a conversation starts, each incoming user act is matched with each state partition. If there is no match, the system consults the ontology rules and creates a match by expanding the tree.

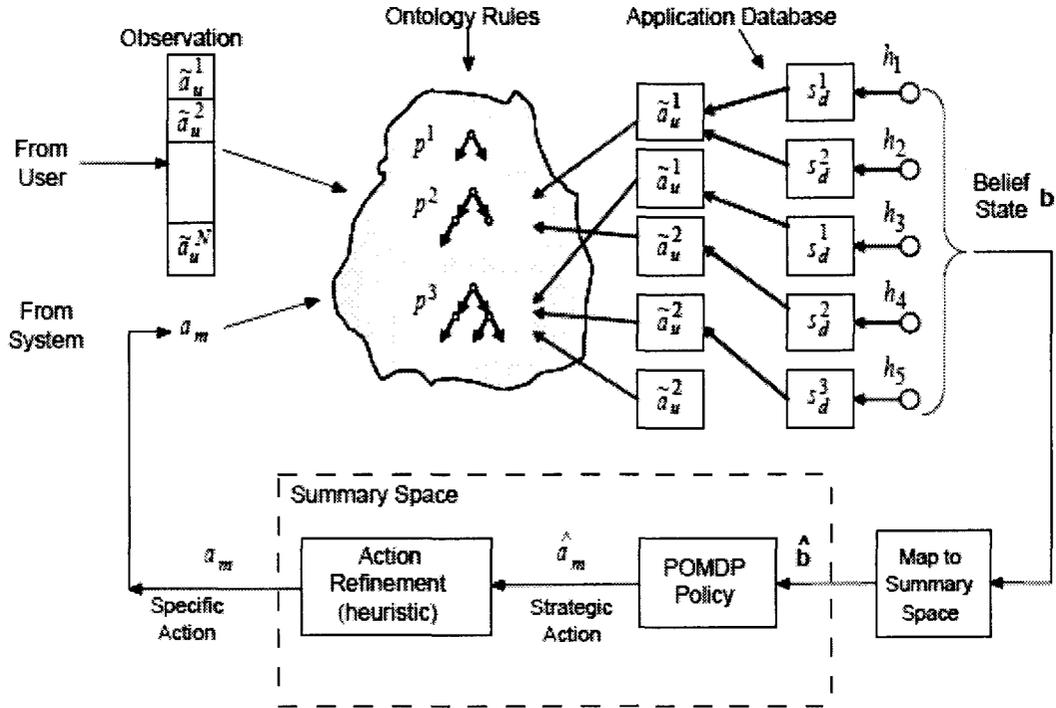


Figure 3.13: HIS Dialogue Manager

In 2008, Young et al. developed a framework to evaluate the HIS dialog manager in noisy environment [Gasic08] and for modelling user behaviour [Keizer08]. In first case, they used HIS dialogue manager as example and MDP – based dialogue manager as baseline to evaluate the results for both simulated and real environment for the tourist information domain. The results proved that the HIS dialogue manager in real environment performed better than MDP based dialogue manager, whereas in simulated environment it fails to model the uncertainty which made POMDP model to exploit alternative hypothesis for spoken dialogue systems. In later case, they designed a User Act Model (UAM) as part of HIS to evaluate the user behaviour in the spoken dialogue systems in noisy environment. This system proved its robustness at higher error rates, providing good quality N-best are provided.

### **3.1.6 Limitations of existing POMDP Approaches**

Though POMDP based dialog managers are used to develop lot of spoken dialogue systems, the questions to be answered and issues to be solved remains the same. It considers the real world state as static and the decision made by POMDP depends on the current belief state. The dialogue manager consists of a component to update the dialogue history, but those dialogue histories are not considered for making effective decisions. The issue of handling uncertainty remains the same and continues till date in all existing approaches of POMDP dialogue manager.

### **3.2 Review of Contextual Control Model**

Hollnagel in 1993 developed a Contextual Control Model (COCOM) to control and analyse team behaviour 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 behaviour should be analysed as macro rather than micro level. These control modes of team behaviour varies in terms of forward planning.

#### **3.2.1 Testing COCOM by Assessing Team Behaviour**

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 behaviour* could be categorised 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 [Stanton01]. 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.

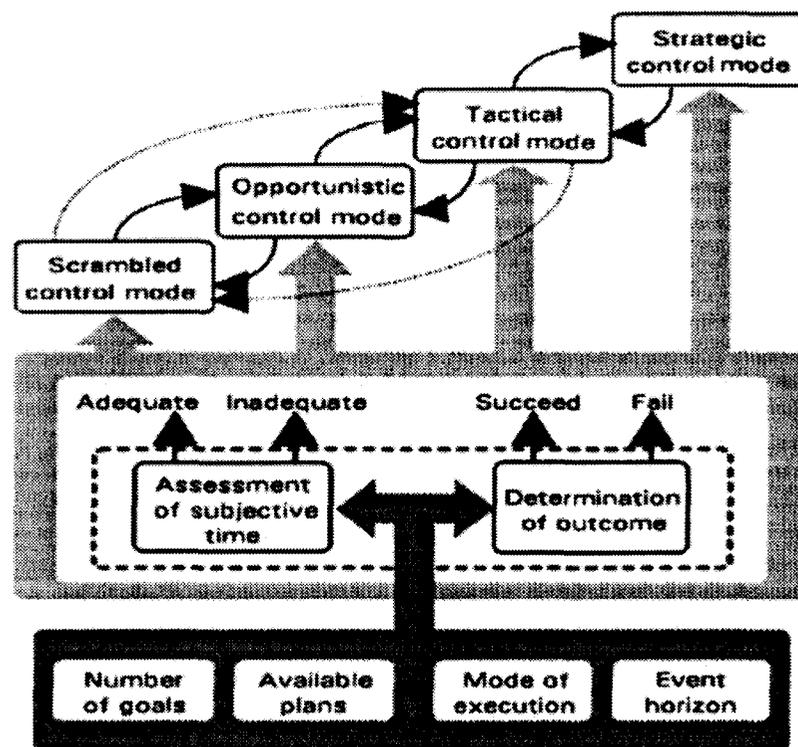


Figure 3.14: Internal structure of Contextual Control Model [COCOM]

They tested the method with 24 groups of people, 4 in each group who has some experience or interest in engineering background. There were 74 males and 22 females of the age between 19 and 55. The study tested between factors using four different conditions,

where six teams of four people were asked to perform a simulated task of balancing a gas-network system. The dependent variable measures were time spent in each control mode by each team and transitions between control modes [Stanton01]. Four networked pc's were used for experiments, video cameras to capture the behaviour and telephones for communication between the team members. The data analysis was carried out using Mann Whitney U test to check any statistical difference between the groups. They also analysed a transition from scrambled to strategic mode. The results showed that the teams spend more time on tactical mode supporting Hollnagel's COCOM.

In 2004, Eric Hollnagel developed a framework to use COCOM in human reliability analysis known as Cognitive Reliability and Error Analysis Method (CREAM) to model the human performance as a set of control modes. The transition between modes depends on the following strategy; the strategic mode based on long-term planning, tactical mode based on set of procedures, opportunistic mode based on the present context and the scrambled mode based on random response. In 2006, Kim et al., proposed a probabilistic method namely Bayesian network for determining the control modes [Kim06]. Renner and Johansson used the COCOM in traffic environment to test the single driver behaviour, coordination and interaction between drivers in traffic environment [Renner06].

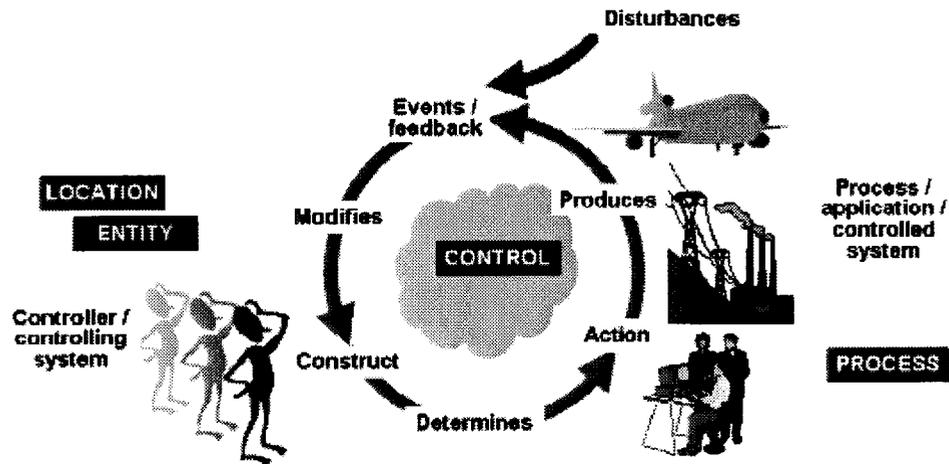


Figure 3.15: Contextual Control Model by Eric Hollnagel in 1998 [Renner06]

To accomplish this work, they proposed a framework called ECOM which is nothing but the extended version of COCOM. ECOM provides a framework for analysing single driver behaviour aiming at both lower- level and higher-level goals. While considering about coordination between drivers on the road or intersection each driver makes an assumption on other driver's intention based on traffic rules.

### 3.2.2 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 individuals transition between COCOM control 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 behaviours which they

determine using this model namely, perception, situation assessment, communication, coordination, analysis, alternative generation and comparison of alternatives and tracked how these behaviours changes under different contexts. Traditionally, support systems are designed to use single human activity, decision making and ignores several behaviours 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 [Feigh06]. 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 [Karen06]. 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.

### **3.3 Conclusion**

In this chapter, we discussed briefly about the proposed works of Partially Observable Markov Decision Process and Contextual Control Modes from literature. From the literature, it is evident that the issues and limitations of both the research areas are still remain unsolved. As the coin has two sides, each approach has its own drawback to be taken care of in future.

## **Chapter 4**

### **Four Mode Based Dialogue Management**

This chapter gives a detailed explanation about the contribution of this thesis. First, the overview of the method description is presented. Second, we have integrated COCOM to spoken dialogue systems for decision making purpose. Then, we discussed how the modified POMDP chooses best action to respond the user's requirement, as the system is dominated by COCOM for control and decision. Finally, the flow chart and pseudo code of the modified approach is presented.

#### **4.1 Method Description**

The previous chapter provided a clear idea about how POMDP can be incorporated with dialogue management to get the actual intention of the human user. There are several techniques have been proposed from finite state based approach to hidden information state, aiming at providing more interaction between the human and the computer. 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 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 natural interaction between the human and machine, it is always wise 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 incorporating Contextual Control Modes for dynamic decision making mechanism. We have also modified the reward model depending upon the four control modes.

We have made *two contributions* in this thesis. First, we have incorporated Contextual Control Modes in Dialogue Management for handling decision making mechanism in the dialogue manager. Second, we have modified the mathematical evaluation of POMDP model depending upon the framework proposed.

#### **4.2 COCOM for Decision Making in DM**

Though POMDP based spoken dialogue systems provide better outcomes in lot of applications, still the researchers put lot of their effort on the decision making part of POMDP as it choose the actions based on the current dialogue state. This can be used in applications which can provide services based on short time goals. On the other hand, to make a system to behave and provide required services in a natural way, it has to react to the situation and not deal with pre-defined words. Theoretically speaking, it is a very tedious effort to develop a system which has the ability to control over the situation and within the time availability, even which is not hundred percent successful with the intervention of humans. To accomplish this type of system, we have integrated Hollnagel's four control modes to the POMDP model, as the dialogue controlling mechanism is taken care of by these modes. Four mode based system choose the best action by the context of the dialogue and

switch between the modes depends on the dialogue states, current action, context of the situation and available time. The system plans what to perform, how to react to the situations depends on the users and the environment. Switching between the four control modes makes the systems reliable and provides services exactly what the user wants. The system maintains its mode in strategy level depends on the context of the dialogue and time availability, aiming at providing higher level goal. If the system is in scrambled mode, which means a panic situation occurred and the system lost its control on the dialogue. In this case, the system is allowed to take random decisions depends on the context of the situation. If a user failed to provide any information, instead of throwing errors or repeating the same query, here the system decides to change its mode to opportunistic and gives options to the user in which he has select the information from the list. So that, the user won't repeat the same mistake in the next dialogue state which increases the timestamps and time to achieve the optimal goal. Sometimes, repeating same questions (pre-set) to the user annoys the user and switches the system into scrambled mode. So, it is always wise to plan what to do in short-term within the time- horizon by considering all the factors before choosing an action rather than repeating the same pre-set of queries. Because the situation in an environment does not remain the same as it varies according to the type of users and the services requested by them in a dialogue management system. By Hollnagel's hypothesis, these four modes are designed to carry out forward planning depending the control, competence and constructs [COCOM].

### **4.3 Modified POMDP Approach**

Unlike, previous POMDP approaches our method provides the services considering all factors from the perspective of the user to achieve the optimal goal with few dialogue states with short time. We have also concentrate more on pruning the number of dialogue states at

least by ten percent depending upon the type of domain. Our dialogue manager has the same components exactly like the previous dialogue managers composed of such as knowledge base, updated dialogue history, discourse generator and session model.

The system starts with the greet message followed by the system query to request for what type service to be provided to the user. Initially, at time  $t$  the system is normally in some unobserved state,  $s \in S$ . When the conversation established between the user and the system, the dialogue states transition from  $s$  to  $s'$  by the increment of time stamps. Choosing the best action for the dialogue in our system is dominated by the four control modes in our system. The decision making or the switch between the control modes depends on the time available to make decisions on the particular context of the dialogue. So, we have introduced a factor  $T_A$  which represents the available time for choosing the best action which depends on the machine state  $s_m$ , set of observations  $o'$ , machine actions  $a_m$  and the belief state  $b(s)$  of the machine, is denoted by  $T_A(s_m, a_m, o', b)$  at each time  $t$ . Depending upon the values of  $T_A$ , and the machine states  $s_m$  the switching between the modes takes place, which does not mean that the decision is taken now. The decision making is done by comparing the state of the system  $s_m$ , machine actions  $a_m$ , observation  $o'$ , belief state  $b$  and the type of control mode  $T_A$  at present the dialogue is in (i.e. at time  $t$ ). To calculate the belief distribution of the dialogue, we have also introduced a factor  $t_r$  which represents the response time of the system to the user in milliseconds. So, the belief state distribution is updated based on  $o'$  and action  $a$  are as follows:

$$\begin{aligned}
 b'(s') &= P(s' | o', a, b, t_r) \\
 &= \frac{P(o' | s', a, b, t_r) P(s' | a, b) P(s' | a, t_r)}{P(o' | a, b, t_r)}
 \end{aligned}$$

$$\begin{aligned}
&= \frac{P(o' | s', a) \sum_{s \in S} P(s' | a, b, t_r, s) P(s | a, b) P(s | a, t_r)}{P(o' | a, b, t_r)} \\
&= \frac{P(o' | s', a) \sum_{s \in S} P(s' | a, s) b(s) t_r(s)}{P(o' | a, b, t_r)}
\end{aligned}$$

Here, the dialogue states and actions represent the machine states and machine actions respectively. And the value of  $T_A$  depends on the current action and belief state distribution, which is given by  $T_A(o' | s, a, b')$ . Based on the current belief state and available time, the machine selects an action  $a \in A$ , receives an reward  $r(s_m, a_m)$  and transitions to a new unobserved state  $s'$ . Then, the system receives an observation  $o' \in O$  depends on the system state  $s_m$  and the system action  $a_m$ . Finally, the belief state is updated with a new one at particular time  $t$ . In our system, the belief state value depends on the fields: type of crust, pizza size, number of pizza, pizza toppings.

#### 4.3.1 Rewards

We have also changed the reward model depending upon the modified POMDP approach with four control modes. Previous POMDP model, has two types of reward with some positive values for correct dialogue, i.e., the system exactly understand the user utterances and provide service exactly what the user wants. Negative values or zero for incorrect dialogue in which the system does not understand the user utterances exactly. In our model, we have modelled the rewards type depending upon the control modes: +100 if the dialogue state is in strategic mode and tactical mode as the system understands what the user wants, -100 if the dialogue is in opportunistic mode as the user doesn't provide correct information or if there any conflict in the information provided by the user. Here, the system receives a negative reward and instructed to provide options to the user to provide interactive dialogue

between the user and the system. Zero if the system is in scrambled mode because in this mode, the user doesn't receive any proper information or query from the user which may be some disturbances, error or some corrupted information.  $\lambda$  is used as discount factor at time  $t$ , and the reward  $R$  is given by,

$$R = \sum_{t=0}^{\infty} \lambda^t r(b_t, a_{m,t}, t_r) = \sum_{t=0}^{\infty} \lambda^t \sum_{s \in S} b_t(s) t_r(s) r(s, a_{m,t})$$

Each action is determined by a policy  $\pi$  and POMDP system involves in finding the optimal policy  $\pi^*$  for the application which maximizes the rewards.

$$\pi^*(s_t) = a[\text{argmax}_{a \in A}(b_t, t_r)]$$

### 4.3.2 Confidence Scores

We have also incorporated the confidence score by providing an estimation of real value to show how exactly that the system understands the user utterances denoted by  $c$  with a pre-defined threshold value 0, which in turn affects the rewards received for each dialogue state.

We haven't made any changes to the confidence buckets as it depends on the user's utterances and system observation denoted as user action  $a_u$ . So, we have used the same evaluation for calculating confidence score as it doesn't make any change choosing system actions. But we have included an option of paraphrasing or double checking mechanism to increase the confidence scores. In this case, the system transitions to tactical mode because at this case the system reconfirms the user utterance by providing him an option in order to understand the user requirement exactly and reach the optimal goal within the available time.

For example, consider this dialogue for Pizza – ordering domain [William05]. If the user asks for normal size pizza, the system here gets confused with the size field “normal”, and here the confidence score will be low or the application assumes the size with the pre-defined

values for the size field and continues the dialogue. Finally, the user goal will not be reached which may switch the system to panic situation or deliver the user with wrong size of pizza.

S: And what type of crust?  
U: Uh just normal  
[large normal]  
Confidence score: 0.35

```
order: {  
  size: large [?]  
  confidence: low [?]  
}
```

But our model, transitions to the tactical mode and double checks with the option we have pre-defined in the size field.

*S: What type of crust?*

*U: Regular*

*S: Thank you. So you need a Regular (12") pizza?*

In this case, the confidence score is 0.63 and the system understands the user's requirement exactly in one time step. So, the optimal goal will be reached in few time steps which makes the system more reliable and less recognition error. If the environment is noisy or lot of speech error, the system is reported to be in scrambled mode. Then it is instructed to transition to change in to opportunistic mode, so that the system can display or tell the options to the user to select from the list. The change in modes depends on the user action, availability time and the rewards received. The developed dialogue manager also handles if there any conflict in the dialogue like, if the user request for vegetarian pizza with the toppings cheese, green olives and pepperoni which shows a conflict with the type field. Since there is a conflict, the belief state value for topping will be 0 and the confidence score will be low and rewards received will be in negative which makes the systems to transitions to opportunistic mode and provides available toppings for the vegetarian pizza.

A graphical representation of our framework and its pseudo code is also given as follows. In the pseudo code and flow chart, M represents the four control modes namely, X – Scrambled mode, O- Opportunistic mode, T- Tactical mode and S- Strategic mode.

---

### **Pseudo – Code of Proposed Model**

---

- 1. function PIZZAORDERINGDOMAIN()**
2.  $t \rightarrow$  initialize time- stamp 0
3.  $s_m \rightarrow$  initialize system state (unobserved)
4.  $a_m \rightarrow$  initialize system action
5.  $b \rightarrow$  initialize belief state 0
6.  $M \rightarrow$  initialize control mode 0
- 7. repeat**
8.  $s_u \rightarrow$  user dialogue state
9.  $a_u \rightarrow$  user dialogue acts  $\pi(s_u)$
10. Calculate belief-state  $b(s)$
11. Calculate Confidence Score C
- 12. Calculate system response time  $T_r$**
- 13. Check mode  $M(X, O, T, S)$**
14. Generate machine action,  $a_m =$  dialogue act type  $a_m$  in the context
- 15. Generate Rewards  $r(s,a,t_r)$**
16. If machine action,  $a_m \neq$  dialogue act type  $a_m$  in the context
- 17. Change mode  $M(X, O, T, S)$**
- 18. Update belief-state  $b(s)$**
- 19.  $t \rightarrow t+1$**
20. **until** the conversation terminates
- 21. end function**

Table 4.1 Pseudo – Code of Proposed Method

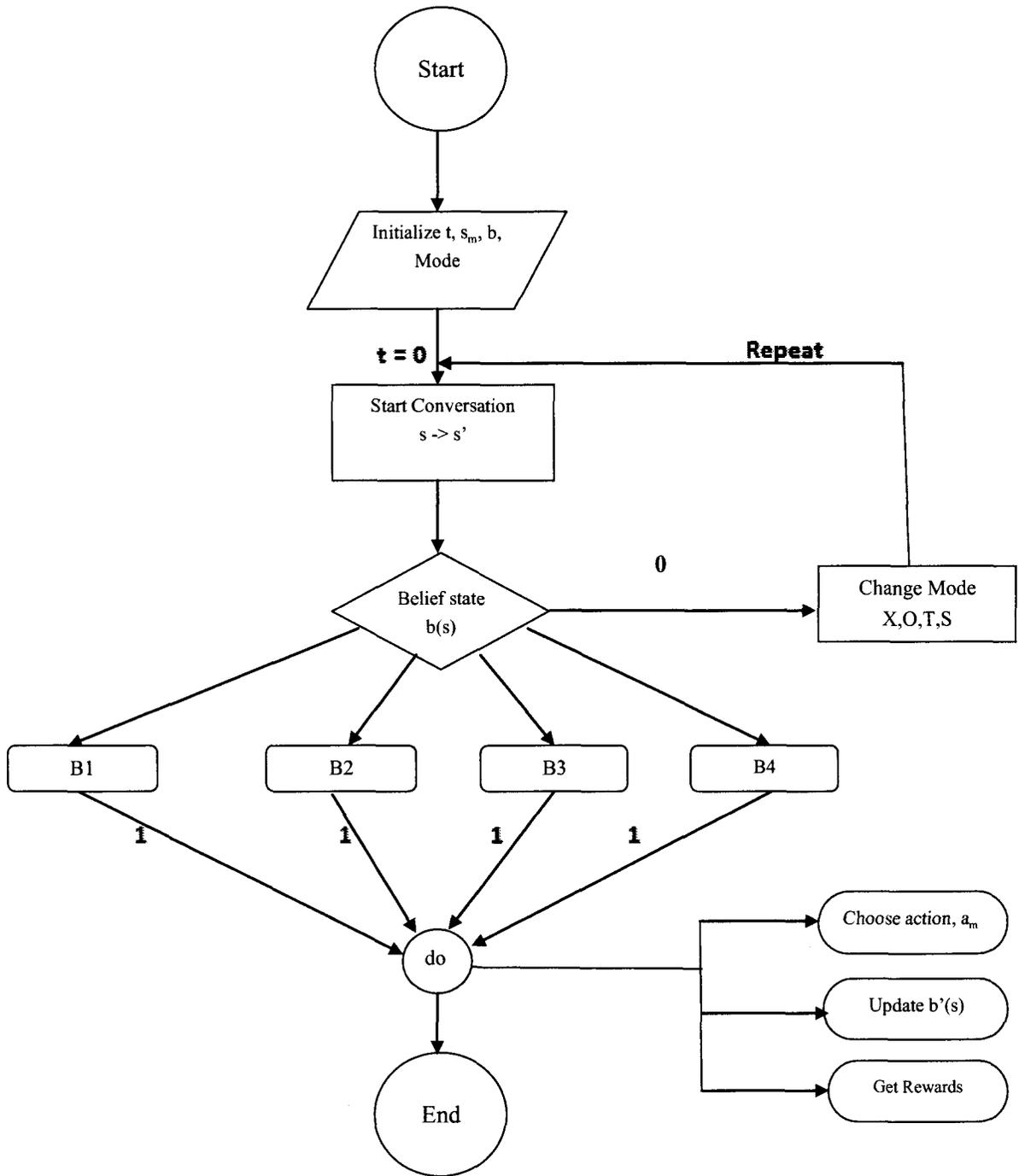


Figure 4.1: Flow-Chart of Proposed Model

## **4.4 Conclusion**

In this chapter, we have discussed about the contributions made in the POMDP based dialogue management systems to make dynamic decision making depending on the four control modes. We have also presented the modified approach of POMDP for handling real-world state and uncertainty. Also, we have discussed how our approach extends the reward model and confidence scores. The main advantage of our proposed model is the dynamism and robustness compared to the different dialogue management system approaches.

## **Chapter 5**

### **Experimental Analysis**

This chapter describes about the implementation of our proposed system such as software and hardware requirements. Then, the tools used to develop and test the system followed by the dialogue manager developed to carry out experiments with an example domain. Finally discussions about the experimental results are presented.

#### **5.1 Implementation**

Since we are using two software modules; one for user output and other one for knowledge base we need a computer with 2 GB RAM, Pentium Dual core with 2 GHz processor and the system should be connected to a mike and speaker. The proposed system is implemented using JAVA under Eclipse 3.5 and the knowledge base has been designed using MS SQL Server and a connection has been established between both front – end and back – end applications. We have also used an audio system known as Mary TTS system which is also started with the execution of dialogue manager in order to provide speech input and output for the system. It synthesis the input and output data in wave file format. It provides voice input to the system and generates the speech format output to the user. Figure 16, shows the actual output display of our proposed system with an example between the human user and the system.

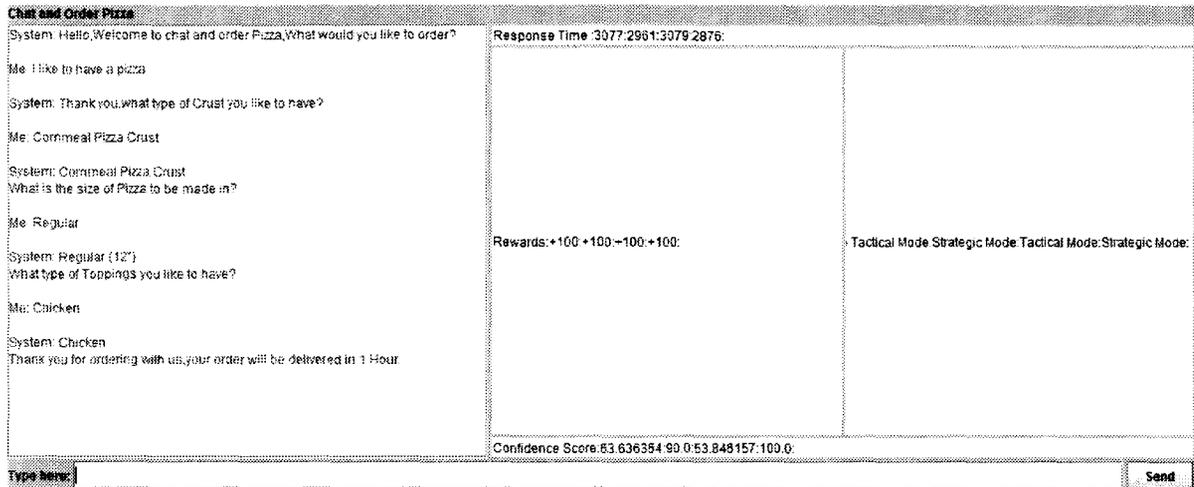


Figure 5.1: User Interface of Proposed system

In the user interface/chat screen, we have also displayed the response time, rewards, mode of the dialogue and confidence scores just for our own tracking purpose to test whether the system performs efficiently. As we use the history of the dialogue to make decisions or choose system actions, the developed dialogue manager tracks the system actions, confidence scores, rewards, mode of the dialogue and transition between the modes, response time of the system for every dialogue states and the belief states. The following figure17, shows the dialogue manager of our proposed system. The results collected at the end of each dialogue.

```

##### dialog manager
#####User Type:      System Action:
Rewards:      Confidence Score:      Mode:      Response Time      S
welcomeMessage 0      0.0      OU      userQuestion      -100      25.0
Opportunistic Mode      21415      ask userQuestion      0      0.0      OU
userQuestion      100      63.636364      Tactical Mode      10005      ask crustType      0
0.0      OU      crustType      -100      25.0      Opportunistic Mode
1735S      ask crustType      0      0.0      OU      crustType      100
65.21739      Tactical Mode      1265S      ask pizzaSize      0      0.0      OU
pizzaSize      100      53.846157      Tactical Mode      12345      ask pizzaTopping
0      0.0      OU      pizzaTopping      100      100.0      Strategic Mode
1063S      ask userQuestion      0      0.0      0
##### Response Time
#####1280675816890-0.0:1280675816921
-0.0:1280675831781-42.857143:1280675831812-33.333336:1280675831843-
50.0:1280675831875-63.636364:1280675843265-0.0:1280675843281-0.0:1280675854781-
25.0:1280675854796-0.0:1280675854812-0.0:1280675863031-53.846157:1280675870468-
0.0:1280675870484-0.0:##### Round, Belief State
entry,values #####Round#: Belief
State:(crustType-pizzaTopping-pizzaSize)      value: 0      0-0-0      0.0      1
0-0-0      0.0      2      0-0-0      0.083333336      3      1-0-0      0.21739131      4
1-1-0      0.3968785      5      1-1-1      0.73021185

```

Figure 5.2: Dialogue Manager updates all the activities of the system

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 analyse the results of the POMDP dialogue manager [Tool07]. We have specified our problem in a specification POMDP file which is included in the toolkit. 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.

## 5.2 Results

We have tested our proposed system using Pizza –Ordering domain. We have done several experiments to test our system on considering four cases. 1) Normal dialogue between the user and the system, 2) With speech and text errors, 3) To handle conflicts in the user requirement, 4) To check whether the system updates the dialogue states and provide services depends on the history of the dialogue.

The following figures show the result for the first case which is the normal dialogue between the human user and the system followed by the belief state values and transition in modes are illustrated. After each dialogue, the updated history of dialogue manager is viewed. Several experiments have been done and the belief state values and change in modes are presented as follows. Here, the results are arranged to show exact difference in values.

The screenshot displays a chat window titled "Chat and Order Pizza". The dialogue is as follows:

- System: Hello, Welcome to chat and order Pizza. What would you like to order?
- Me: I like to have a pizza
- System: Thank you, what type of Crust you like to have?
- Me: Cornmeal Pizza Crust
- System: Cornmeal Pizza Crust  
What is the size of Pizza to be made in?
- Me: Regular
- System: Regular (12")  
What type of Toppings you like to have?
- Me: Chicken
- System: Chicken  
Thank you for ordering with us, your order will be delivered in 1 Hour

On the right side of the chat window, there is a panel with the following information:

- Response Time : 3077.29613079:2876:
- Rewards \*100 \*100 \*100 \*100:
- Tactical Mode: Strategic Mode Tactical Mode: Strategic Mode:
- Confidence Score E3.636554:90 0:59.646157:100 0:

At the bottom of the chat window, there is a text input field labeled "Type here:" and a "Send" button.

Figure 5.3: User Interface for case 1 experiment

```

##### Chat Log
#####
User Type:      System Action:  Rewards:      Confidence Score:  Mode:
Response Time
S    welcomeMessage  0          0.0          0
U    userQuestion    100        71.42857     Tactical Mode    2946
S    ask crustType   0          0.0          0
U    crustType       100        86.95652     Strategic Mode   2892
S    ask pizzaSize   0          0.0          0
U    pizzaSize       100        53.846157    Tactical Mode    3038
S    ask pizzaTopping 0          0.0          0.0            0
U    pizzaTopping    100        100.0        Strategic Mode   2768
S    endQuestion     0          0.0          0
##### Response Time
#####1282640573171-
21.428572:1282640573176-27.272728:1282640573180-71.42857:1282640573183-3.7037036:
1282640593845-86.95652:1282640593857-47.61905:1282640593867-50.0:1282640593877-
0.0:1282640593887-38.46154:1282640593896-0.0:1282640593906-0.0:1282640593916-
0.0:1282640593926-27.777779:1282640614148-0.0:1282640614149-0.0:1282640614150-
53.846157:1282640622589-0.0:1282640622590-0.0:1282640622591-0.0:1282640622592-
0.0:1282640622593-0.0:1282640622594-0.0:1282640622595-0.0:1282640622596-
0.0:1282640622597-0.0:1282640622598-0.0:1282640622599-0.0:1282640622600-
0.0:1282640622601-0.0:1282640622602-0.0:1282640622603-0.0:1282640622604-
0.0:1282640622605-0.0:##### Round, Belief State
entry, values ##### Round#: Belief
state:(crustType-pizzasize-pizzaNos-pizzaTopping) conflict value:
1      0-0-0-0 N      0.0
      1-0-0-0 N      0.2173913
      1-1-0-0 N      0.35200667
      1-1-0-1 N      0.6020067

```

Figure 5.4: The updated history component of proposed dialogue manager for case 1

Case 2: The experiment shows how the dialogue manager reacts to speech or text errors.

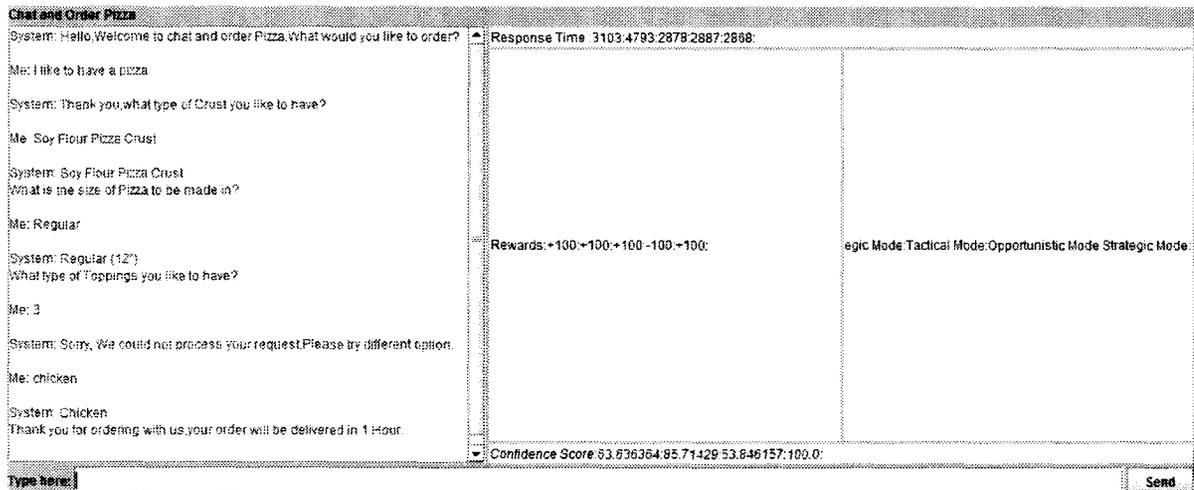


Figure 5.5: User interface for case 2 experiments

```

***** Chat Log *****
User Type:      System Action:  Rewards:      Confidence Score:  Mode:  Response Time
S              welcomeMessage  0             0.0           0
U              userQuestion  100          63.636364    Tactical Mode    3103
S              ask crustType  0            0.0           0
U              crustType    100          85.71429     Strategic Mode    4793
S              ask pizzasize  0            0.0           0
U              pizzasize    100          53.846157    Tactical Mode    2878
S              ask pizzaTopping  0            0.0           0
U              pizzaTopping -100         25.0         Opportunistic Mode 2887
S              ask pizzaTopping  0            0.0           0
U              pizzaTopping  100          100.0        Strategic Mode    2868
S              enduserQuestion 0            0.0           0
***** Response Time *****
1282638198638-50.0:1282638198662-63.636364:1282638198670
-42.857143:1282638198679-33.333336:1282638214652-43.47826:1282638214731-85.71429:1282638214816-
50.0:1282638214896-0.0:1282638215088-38.46154:1282638216050-
0.0:1282638216128-0.0:1282638216207-0.0:1282638216288-27.777779:1282638225030-0.0:1282638225031-
0.0:1282638225032-53.846157:1282638281677-0.0:1282638281678-
0.0:1282638281679-0.0:1282638281680-0.0:1282638281681-0.0:1282638281682-0.0:1282638281683-
0.0:1282638281684-0.0:1282638281685-0.0:1282638281686-0.0:1282638281687-
0.0:1282638281688-0.0:1282638297297-0.0:1282638297298-0.0:1282638297299-0.0:1282638297300-
0.0:1282638297301-100.0:1282638297302-0.0:1282638297303-0.0:1282638297304-
0.0:1282638297305-0.0:1282638297306-0.0:1282638297307-0.0:1282638297308-0.0:1282638297309-
0.0:1282638297310-0.0:1282638297311-0.0:1282638297312-0.0:
***** Round, Belief State
entry, values*****
Round#: Belief State:(crustType-pizzaSize-pizzanos-pizzaTopping)  conflict  value:
1      0-0-0-0      N      0.0
      1-0-0-0      N      0.21428572
      1-1-0-0      N      0.3489011
      1-1-0-0      N      0.4114011
      1-1-0-1      N      0.5989011
      0-0-0-0      N      0.0

```

Figure 5.6: The updated history component of proposed dialogue manager for case 2

Case 3 and 4: Handling conflicts in user information and checking updated history for choosing system actions.

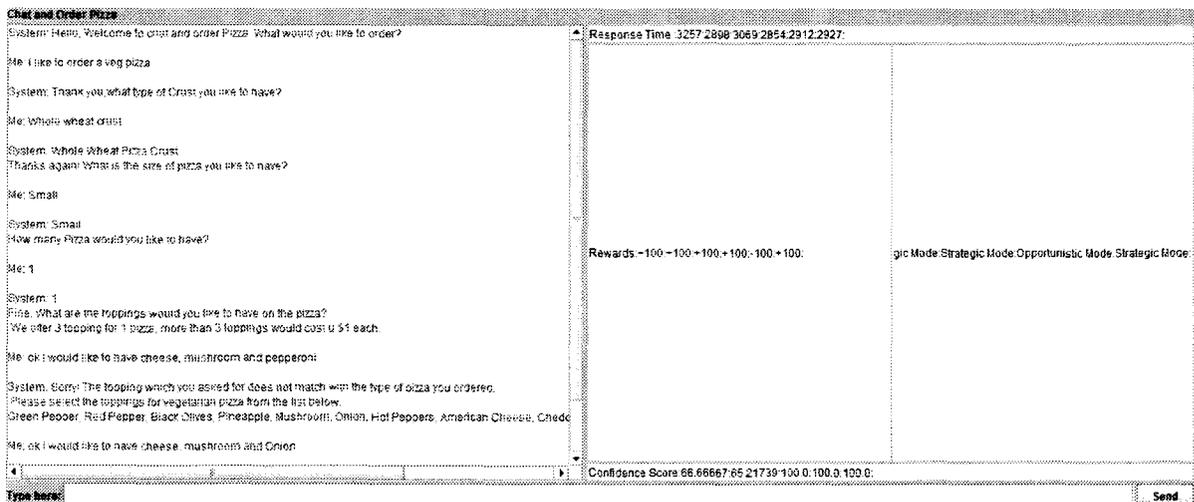


Figure 5.7: User Interface for case 3 and 4 experiment

```

##### Chat Log #####User
Type: System Action: Rewards: Confidence Score: Mode: Response Time S
welcomeMessage 0 0.0 0U userQuestion 100 66.66667 Tactical Mode
32575 ask crustType 0 0.0 0U crustType 100 65.21739
Tactical Mode 28985 ask pizzaSize 0 0.0 0U pizzaSize 100 100.0
Strategic Mode 30695 ask pizzaNos 0 0.0 0U pizzaNos 100 100.0
Strategic Mode 28545 ask pizzavegTopping 0 0.0 0U pizzavegTopping -100
25.0 Opportunistic Mode 29125 ask pizzavegTopping 0 0.0 0U
pizzavegTopping 100 100.0 Strategic Mode 29275 ask userQuestion 0 0.0
##### Response Time #####
66.66667:1282643303798-65.21739:1282643303827-18.518518:1282643316345-65.21739:1282643316347-
23.809525:1282643316349-25.0:1282643316351-0.0:1282643316353-38.46154:1282643316355-
0.0:1282643316357-0.0:1282643316358-0.0:1282643316360-27.777779:1282643333763-0.0:1282643333764-0.0:
1282643343294-0.0:1282643343295-0.0:1282643400617-0.0:1282643400618-0.0:1282643400619-
40.0:1282643400620-0.0:1282643400621-0.0:1282643400622-0.0:1282643400623-0.0:1282643400624-
0.0:1282643400625-0.0:1282643400626-0.0:1282643400627-0.0:1282643400629-0.0:1282643400630-
0.0:1282643400633-0.0:1282643400634-0.0:1282643400636-0.0:1282643400637-0.0:1282643400638-
0.0:1282643400640-0.0:1282643400641-0.0:1282643400642-0.0:1282643400643-0.0:1282643435139-
0.0:1282643435140-0.0:1282643435141-0.0:1282643435142-0.0:1282643435143-0.0:1282643435144-
0.0:1282643435145-0.0:1282643435146-0.0:1282643435147-0.0:1282643435149-0.0:1282643435150-
0.0:1282643435151-0.0:1282643435152-0.0:1282643435153-0.0:1282643435154-100.0:1282643435155-
0.0:1282643435157-0.0:1282643435158-0.0:1282643435159-0.0:1282643435160-0.0:1282643435161-0.0:
##### Round, Belief State entry, values #####
#####Round#: Belief State:(crustType-pizzaSize-
pizzaNos-pizzaTopping) Conflict value:
1 0-0-0-0 N 0.0
1-0-0-0 N 0.16304348
1-1-0-0 N 0.41304347
1-1-1-0 N 0.6630435
1-1-1-0 Y 0.1255435
1-1-1-1 N 0.9130435

```

Figure 5.8: The updated history component of proposed dialogue manager for case 3 and 4

We have collected the results for belief state values for each experiment which have been evaluated using our modified POMDP equation. The value 1 represents that the user gave exact information for the particular field. The value 0 represents the user's information for that field is not provided or conflict with the type of field.

Round#	Belief State:(crustType-pizzaSize-pizzaNos-pizzaTopping)	Conflict	Value
0	0-0-0-0	N	0.0
	1-0-0-0	N	0.16304348
	1-1-0-0	N	0.41304347
	1-1-1-0	N	0.6630435
	1-1-1-1	N	0.9130435
1	0-0-0-0	N	0.0
	1-0-0-0	N	0.0187823
	1-1-0-0	Y	0.4285967
	1-1-0-0	N	0.7903435
	1-1-1-0	N	0.8530435
2	1-1-1-1	N	0.9130435
	0-0-0-0	N	0.0
	1-0-0-0	N	0.21473942
	1-0-1-0	N	0.4892017
	1-1-0-0	N	0.6630435
3	0-1-1-0	N	0.6983019
	1-1-1-0	N	0.7356290
	1-1-1-1	N	0.9557392
	0-0-0-0	N	0.0
	1-0-0-0	N	0.1378272
5	1-1-0-0	N	0.2477291
	1-1-1-0	N	0.57537914
	1-1-1-0	N	0.7535221
	1-1-1-1	N	0.93523917
	0-0-0-0	N	0.0
6	1-0-0-0	N	0.0132719
	1-1-0-0	N	0.3467238
	1-0-1-0	Y	0.05903435
	1-1-1-0	N	0.67201734
	1-1-1-1	N	0.90271872
6	1-1-1-1	N	0.92762934
	0-0-0-0	N	0.0
	1-0-0-0	N	0.1026389
	1-1-0-0	N	0.5678261
	1-1-1-0	N	0.5921671
	1-1-1-0	N	0.6391269
	1-1-1-1	N	0.9695382
	1-1-1-0	Y	0.1618185
1-1-1-0	N	0.5692827	
1-1-1-1	N	0.9685719	

Table 5.1 Experimental Results of Modified POMDP model

### 5.3 Discussion

*Case 1:* A normal dialogue between the user and the system in pizza – ordering 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. The component also tracks all possible values

required to make decision making and choose best action to respond to the user. The following figure shows the results and values updated at each dialogue state. The values are arranged to show clearly the system actions at each time steps, belief state values and transition between control modes. To track the rewards and transition mode for each dialogue state, we have also displayed in the user interface for our experimental purpose.

*Case 2:* The experiment shows how the dialogue manager reacts to speech or text errors. The system receives a negative reward as it is affected by noisy environment or text errors. Then it changes its mode to opportunistic and gives some options to the user.

*Case 3 and 4:* Handling conflicts in user information and checking updated history for choosing system actions. If the user order for vegetarian pizza and requests for a non-vegetarian topping results in conflict between the toppings field of the domain. The conflict in the values gives the system a negative reward and transitions from scrambled mode to opportunistic mode and provides set of options for the user to choose from the list. This is also evident that the dialogue manager updates its dialogue history and considers it before choosing system actions. The “Y” in the results shows that there is a conflict in the fields.

## **5.4 Conclusion**

We have showed the exact results of our proposed POMDP framework using four modes for controlling the dialogue. We have also discussed about the limitations of existing dialogue manager which has been overcome by our proposed model which is evident from the above results.

## Chapter 6

### Conclusions and Future Work

Pomdp based dialogue management systems has got more attention in the field of human-computer interaction between the researchers all around the world. To design and implement a dynamic framework for this model is a complex work. Because, if we have simple domain then we can design and maintain a dynamic system and it will be cost effective. But recent applications of spoken dialogue management have been incorporated Artificial Intelligence field to implement a dynamic POMDP – based dialogue management system. As spoken dialogue management applications needs good system requirements to implement and maintain. We introduced contextual control model into dialogue management to handle this issue and provide dynamic decision making in the spoken dialogue systems which is impossible with POMDP model alone.

We have discussed the previous approaches and techniques used in the dialogue management along with its limitations. This thesis motivates how to design and develop a dynamic based system and the evaluation techniques we used to accomplish the task. We have also modified the existing POMDP equations according to the proposed system in order to perform according to the specification. Experimental results add more confidence to the implemented system. We have done several experiments based on different cases to overcome the limitations of existing approaches by using pizza- ordering domain as our test bed. The result also proves that our proposed model is effective and performs efficiently by handling real – world certainty.

Our proposed method in future can be used in both the fields such as dialogue management and support systems. During experiments, we predicted minor errors by setting different values for the discount factor. In some set of dialogue, the decision is accurately the same but in one of hundred cases, if the dialogue is in scrambled mode the system never transition to higher level goals. But still it depends on the type of domain the method is applied on.

In future work, we can use both Pomdp and contextual control modes in the field of Artificial Intelligence to track the behaviour of the Robot and control its decision making features. We can extend the system by adding emotions as input and output to the systems. As we all fond of sending emotions in the form of smileys instead of typing the actual text. So, we can design a system to process emotional data and understands user intention from those emotions. Further, the systems can be made domain independent and language independent. It should be developed as a common concept and can be tested with different spoken dialogue system applications. We can further import the system to mobile application by applying mobile computing techniques in the system. These types of developments in the field of dialogue management will dominate the world technology by using avatars and robots to act as more natural way in providing support like humans.

## Bibliography

- [Alshawi01] H Alshawi and S Douglas. Variant transduction: A method for rapid development of interactive spoken interfaces. In Proc SIGdial Workshop on Discourse and dialogue, Aalborg, Denmark, 2001.
- [Brad98] Brad A. Myers. A Brief History of Human Computer Interaction Technology. ACM Interactions. Vol. 5, no. 2, March, 1998. pp. 44-54.
- [Bohus03] D. Bohus and A. Rudnicky. Ravenclaw. Dialog management using hierarchical task decomposition and an expectation agenda. In Eurospeech-2003, Geneva, Switzerland, 2003.
- [Balantine99] B Balantine, DP Morgan, and WS Meisel. How to build a speech recognition Application. Enterprise Integration Group, 1999.
- [Bui06] Bui, T.H. Multimodal Dialogue Management - State of the art. Technical Report TR-CTIT-06-01, Centre for Telematics and Information Technology, University of Twente, Enschede. ISSN 1381-3625, 2006.
- [Bui et. al.07] T.H. Bui, B.W. van Schooten and D.H.W. Hofs. Practical Dialogue Manager Development using POMDPs. In Proceedings of the 8th SIGdial Workshop on Discourse and Dialogue, S. Keizer, H.C. Bunt and T. Paek (eds), The Association for Computational Linguistics, PA, USA, pp. 215-218, 2007.
- [Bui07] T.H. Bui, M. Poel, A. Nijholt and J. Zwiers,. A tractable DDN-POMDP Approach to Affective Dialogue Modeling for General Probabilistic Frame-based Dialogue Systems. In Proceedings of the 5th IJCAI

- Workshop on Knowledge and Reasoning in Practical Dialogue Systems, D. Traum, J. Alexandersson, Jonsson and I. Zukerman (eds), IJCAI, Hyderabad, India, pp. 34-37, 2007.
- [Catizone02] R. Catizone, A. Setzer, and Y. Wilks. State of the Art in Dialogue Management. 2002
- [Churcher97] Gavin E. Churcher, Eric S. Atwell, and Clive Souter. Dialogue management systems: a survey and overview. 1997.
- [Cohen97] P. Cohen. Dialogue Modeling in Survey the State of the art in Human Language Technology. 1997.
- [COCOM] [http://www.ida.liu.se/~eriho/COCOM\\_M.htm](http://www.ida.liu.se/~eriho/COCOM_M.htm)
- [Cuayahuitl09] Heriberto Cuayahuitl, Steve Renals, Oliver Lemon, Hiroshi Shimodaira. Evaluation of a hierarchical reinforcement learning spoken dialogue System. *Computer Speech & Language* 24(2): 395-429, 2010.
- [Denecke02] M Denecke. Rapid prototyping for spoken dialogue systems. In Proc Intl Conference on Computational Linguistics (COLING), Taipei, 2002.
- [Feigh06] 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.
- [Foster02] Mary Ellen Foster. State of the art review: Multimodal Fission. 2002.
- [Gasic08] M. Gasic, S. Keizer, F. Mairesse, J. Schatzmann, B. Thomson, S. Young. Training and Evaluation of the HIS POMDP Dialogue System in Noise. *Sigdial* 2008, Columbus, Ohio, 2008.
- [Goddeau96] D. Goddeau, H. Meng, J. Polifroni, S. Seneff, and S. Busayapongchai. A

- Form-Based Dialogue Manager for Spoken Language Applications. InProc. ICSLP '96, volume 2, pages 701-704, Philadelphia, PA, 1996.
- [Hasen98] EA Hansen. Solving POMDPs by searching in policy space. In Proc Uncertainty in Artificial Intelligence (UAI), Madison, Wisconsin, 1998.
- [Hofs03] 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.
- [Hulstijn96] H. Hulstijn, R. Steetskamp, H. Doest, S. Burgt, and A. Nijholt. Topics in schisma dialogues. 1996.
- [Infriek] <http://www.vf.utwente.nl/~infrieks/dialogue.html>
- [Jason05] 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.
- [Jason08] Jason D. Williams. Exploiting the ASR N-Best by tracking multiple dialog state hypotheses. Proc Interspeech, Brisbane, Australia, 2008.
- [Jods] <http://www.jods.org/>
- [Johnston02] 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.
- [Karen06] Feigh, Karen and Pritchett, Amy. Design of Multi-mode Support Systems

- for Airline Operations. International Conference on Human Computer Interaction in Aviation. Eurisco, September 2006.
- [Keizer08] S. Keizer, M. Gasic, F. Mairesse, B. Thomson, K. Yu, S. Young. Modelling User Behaviour in the HIS-POMDP Dialogue Manager. IEEE Workshop on Spoken Language Technology (SLT08), Goa, India, 2008.
- [Kim06] MC Kim, PH Seong, E Hollnagel. A probabilistic approach for determining the control mode in CREAM. Reliability Engineering & System Safety, Volume 91, Issue 2, Pages 191-199 2006.
- [Larsson20] 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.
- [LarsTrau20] 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.
- [Levin97] E. Levin, R. Pieraccini. A Stochastic Model of Computer-Human Interaction For Learning Dialogue Strategies. Proc. Eurospeech, 1997.
- [Levin20] E. Levin, R. Pieraccini, W. Eckert. A Stochastic Model for Human-Machine Interaction for Learning Dialog Strategies. IEEE Transactions on Speech and Audio Processing, vol.8, no.1, Jan. 2000.
- [McTear02] M. McTear. Spoken dialogue technology: Enabling the conversational

- User Interface. *ACM Computing Survey*, Volume 34, No 1, 2002.
- [Oldsig] <http://old.sigchi.org/cdg/cdg2.html>
- [Oviatt02] Sharon Oviatt. *Multimodal interfaces*. 2002.
- [Peter et al05] Peters, J., Vijayakumar, S., Schaal, S. Natural actor-critic. In: *European Conference on Machine Learning (ECML)*, Springer, pp. 280–291, 2005.
- [Pineau03] J Pineau, G Gordon, and S Thrun. Point-based value iteration: An Anytime algorithm for POMDPs. In *Proc Int. Joint Conference on AI (IJCAI)*, Acapulco, Mexico, 2003, pp. pp1025–1032.
- [Plamen03] Prodanov, Pl., Drygajlo, A. Bayesian networks for Spoken Dialogue Management in multimodal systems of tour-guide robots. *Proceedings of The 8th European Conference on Speech Communication and Technology, Eurospeech*, Geneva, Switzerland, September 2003, pp. 1057-1060, 2003
- [Poupart04] Pascal Poupart and Craig Boutilier. VDCBPI: an Approximate Scalable Algorithm for Large Scale POMDPs. In *Advances in Neural Information Processing Systems 17 (NIPS)*, pages 1081-1088, Vancouver, BC, 2004.
- [Prodanov04] P Prodanov, A Drygajlo. Bayesian Networks for Error Handling Through Multimodality Fusion in Spoken Dialogues with Mobile Robots. *ISCA Tutorial and Research Workshop (ITRW)*, 2004.
- [Ragni06] Ragni, A., “MDP—Modeling in Dialogue Management. *Dialog Systems Course*. pp. 1—19, Spring 2006.

- [Renner06] L Renner, B Johansson. Driver coordination in complex traffic environments. Proceedings of the 13th European conference on Cognitive ergonomics: trust and control in complex socio-technical systems, Zurich, Switzerland, Pages: 35 – 40, 2006.
- [Roy20] Roy, N., Pineau, J., and Thrun, S. Spoken dialogue management using probabilistic reasoning. Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics (ACL-2000), Hong Kong, 2000.
- [Singh02] S. Singh, D. Litman, and M. Kearns. Optimizing dialogue management with reinforcement learning: Experiments with the njfun system. Journal of Artificial Intelligence Research, 2002.
- [Spaan04] Spaan, M. and Vlassis, N. Perseus: Randomized Point-Based Value Iteration for POMDPs. Technical report, Informatics Institute, University of Amsterdam, Netherlands, 2004.
- [Stanton01] 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.
- [Steve etal05] S. Young, J. Williams, J. Schatzmann, M. Stuttle, K. Weilhammer. The Hidden Information State Approach to Dialogue Management. CUED Technical Report CUED/F-INFENG/TR.544, 2005.
- [Susann98] Susann LuperFoy , Dan Loehr , David Duff , Keith Miller , Florence Reeder , Lisa Harper. An architecture for dialogue management,

context tracking, and pragmatic adaptation in spoken dialogue systems. Proceedings of the 17<sup>th</sup> International conference on Computational linguistics, p.794-801, August 10-14, Montreal, Quebec, Canada, 1998.

- [Traum03] David Traum and Staffan Larsson. The information state approach to dialogue Management,” pages 325-353, 2003.
- [Thomson07] B. Thomson, J. Schatzmann, K. Weilhammer, H. Ye, and S. Young. Training a real-world POMDP-based Dialog System. HLT/NAACL Workshop "Bridging the Gap: Academic and Industrial Research in Dialog Technologies", Rochester, 2007.
- [Thomson10] B. Thomson and S. Young. Bayesian update of dialogue state: A POMDP framework for spoken dialogue systems. *Computer Speech And Language*, 24(4): 562-588, 2010.
- [Tool07] <http://wwwhome.ewi.utwente.nl/~hofs/pomdp/index.html>
- [VanZ96] Gert Veldhuijzen van Zanten. Pragmatic interpretation and dialogue management in spoken-language systems. In Proceedings of the 11th Twente Workshop on Language Technology, TWLT 11, pages 81-88, June 1996.
- [William05] J. Williams, P. Poupart, and S. Young. Partially Observable Markov Decision Processes with Continuous Observations for Dialogue Management. SIGDial, Lisbon, 2005.
- [William06] J. Williams, P. Poupart, S. Young. Partially Observable Markov Decision Processes with Continuous Observations for Dialogue

- Management. Recent Trends in Discourse and Dialogue. Eds L. Dybkjaer and W. Minker, Springer. [expanded version of 2005 SIGDial paper], 2006.
- [Xu02] Weiqun Xu, Bo Xu, Taiyi Huang, and Hairong Xia. Bridging the gap Between dialogue management and dialogue models. revue Traitement Automatique desLangues (TAL), 41(2), Paris, 2002.
- [Young20] Young, S.J., 2000. Probabilistic methods in spoken dialogue systems. Philosophical Transactions of the Royal Society (Series A) 358 (1769), 1389– 1402.
- [Young06] S. Young. Using POMDPs for Dialog Management. IEEE/ACL Workshop on Spoken Language Technology, Aruba, 2006.
- [Young07] S. Young, J. Schatzmann, K. Weilhammer and H. Ye. The Hidden Information State Approach to Dialog Management. ICASSP, Honolulu, Hawaii, 2007.

## **Vita Auctoris**

Sabiha Sathulla was born in the year 1985 in Tamil Nadu, India. She received her Bachelor's of Technology degree with the specialization in Information Technology from Anna University, Tamil Nadu, India. She joined the University of Windsor in January 2008 and is currently a candidate for the Master's degree in Computer Science under the supervision of Dr. Xiaobu Yuan.