2010

A modified approach of POMDP-based dialogue management

Libian Bian
University of Windsor

A MODIFIED APPROACH OF POMDP-BASED DIALOGUE MANAGEMENT

by
Libian Bian

A Thesis
Submitted to the Faculty of Graduate Studies
through the School of Computer Science
in Partial Fulfillment of the Requirements for
the Degree of Master of Science at the
University of Windsor

Windsor, Ontario, Canada
2010

Your file Votre référence
ISBN: 978-0-494-70586-5
Our file Notre référence
ISBN: 978-0-494-70586-5

Author's Declaration of Originality

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

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

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

Abstract

This thesis applies the theory of history information space for a thorough study of dialogue management in major approaches, ranging from the classical approach based upon finite state machine to the most recent approach using partially observable Markov decision process (PODMP). While most of the approaches use various techniques to estimate system state, the POMDP-based approach avoids state estimation and uses belief state for decision making. In addition, it provides a mechanism to model uncertainty and allows for error-recovery. PODMP-based dialogue management demonstrates undeniable advantages in the handling of input uncertainty over all the other approaches.

However, applying Markovian over the belief-state space in the current POMDP models causes significant loss of valuable information in dialogue history, leading to untruthful recognition of user's intention. To improve the performance of POMDP-based dialogue management this thesis introduces belief history into the planning process, and uses not only the current but also the previous belief state for the determination of actions. In the new approach, all changes of belief state require a validation with domain constraints, and an invalid change results in a modification to the actions provided by the POMDP solver. Experiments show that this new approach is able to handle uncertainty caused by user's lack of domain knowledge and practical constraints, thus becoming more accurate in intention recognition.

To my dear parents for the encouragement and support of a lifetime.

Acknowledgements

I would like to express my gratitude to all those who gave me support in the process of my master's study. Especially, I want to thank my supervisor, Dr. Xiaobu Yuan, for his patient guidance and enthusiasm he invested in this whole research. His creativity and strong AI knowledge inspire me and light up the road of my study. Moreover, his integrity influences me and make me understand how to be a good researcher. I also want to express my appreciation as well to my thesis committee members, Dr. Wu and Dr. Rieger, who have all generously given valuable suggestions to help me improve this thesis.

Also, I would like to thank all my classmates and friends who supported me in any respect during the completion of the thesis. With their kindly help and encouragement, I'm more confident in the journey of the life.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Robots have a wide application in different areas, including specialized factory robots for packaging or car production and general-purpose robots capable of performing a variety of functions. Recently, dialogue system based robots or virtual assistants have been combined with the expert system to provide more friendly, flexible and accurate services to the human user to achieve particular task or provide social services. When asking for help, however, users still prefer a more natural way of communication that resembles what happens in real life, and expect more precise answers than being forced to make choices from a usually long list of possible answers. While, the real-time interaction between users and agents has created new challenges, and one of them is information exchange that ensures adequate communication for users to converse their goal of required services [16]. The media of communication normally go through the three senses of seeing, hearing, and touching, the format of communication demands for truthful, relevant, clear, and informative dialogue between users and agents. As the development of the speech recognition techniques and natural language processing techniques, human-computer interaction becomes a realistic application. In the domain of the e-commerce, virtual assistant has been designed instead of the sale assistant to help the customer to find the target product[14]. Also, even some commercial applications such as Rogers Communications Inc. computer-telephony integration customer service agent have been developed and applied in daily life to help customers solve their problem anytime, which at some extent they save the labour.

In a speech-based dialogue system, an agent interacts with a user on a turn-by-turn basis [13]. The main purpose of a speech-based dialogue system is to provide an interface between the user and the agent for the agent to understand the need of the user so that adequate services can be provided. The dialogue system, therefore, needs to process the user's spoken input and to recover from errors [22]. A speech-based dialogue system typically includes the components of input, output, and knowledge, plus the core component of dialogue management. Dialogue management simulates the task model in the specific domain. It also processes semantic inputs from fusion, and decides what the agent should do to respond the user's request to fulfill user's goal.

1.1 Problem Statement

Various approaches of dialogue management have been proposed in the last twenty years, including the classical approach based upon finite state machine and the current approach based upon the popular POMDP model. Finite state machine based approach is only suitable to the well structured task and is lack of flexibility. Frame based approach uses a frame to record the information and is more flexible than finite state machine based approach. Bayes network and Markov Decision Process based approach are probabilistic which can solve some uncertainties to some degree but still have drawbacks such as defects in solving observation uncertainties. Although POMDP based approach is the current popular approach, it still has its own problems to be taken care of. Despite its known problem of scalability, the POMDP-based approach demonstrates undeniable advantages in the handling of input uncertainty over other approaches. However, applying the Markovian over the belief-state space in the current POMDP models causes significant loss of valuable information in dialogue history, leading to untruthful recognition of user's intention. In other perspective, the POMDP-based approach only models the user and maintains the knowledge at the control level. However, when a human user lacks the domain knowledge and provides unreasonable information, a POMDP-based approach can only end the dialogue with the task failure. To improve the performance of POMDP-based approach, this thesis introduces a domain knowledge base and belief history into the planning process, and uses not only the current but also the previous belief state for the determination of actions.

1.2 Contributions

In this thesis, two contributions have been made. First, the main different dialogue management approaches have been analyzed under history information space and the problem of each approach has been revealed. Second, with drawing the conclusion that a POMDP-based approach drops some significant information in terms of the history information space theory, the modified POMDP-based dialogue management approach is proposed to handle the uncertainties in the belief state and to improve the accuracy of user's intention recognition. The experiments under 3 scenarios are conducted to evaluate the proposed method. The results of the experiments prove that the concept, and they demonstrate that the proposed method achieves the expected results.

1.3 Organization of the thesis

In the remaining of part this thesis, chapter 2 provides a literature review about the major approaches of dialogue management. Chapter 3 presents first the concept of history information space, and then conducts an analysis of all the major approaches but the POMDP-based approach. Chapter 4 is the core of this paper, which outlines POMDP models, discusses their shortcoming, and presents a new approach of dialogue management. Three types of experiments are presented in chapter 5, whose results show that the new approach is more accurate in the recognition of user intention, thus making agents more attractive and useful when providing services. Finally, chapter 6 ends with conclusions and points out directions for future work.

Chapter 2

Preliminary

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

2.1 Spoken Dialogue System

Spoken dialogue systems have been defined as computer systems with which humans interact on a turn-by-turn basis and in which spoken natural language plays an important part in the communication [13]. [22] identified that the main purpose of a spoken dialogue system is to provide an interface between a human user and a machine usually computer-based application such as a database or expert system. Also Mctear identified that the main tasks of the dialogue system include processing the user's input and recovering from the errors. Based on this, Mactear categorized different dialogue strategies into three types: finite sate or graph based approach, frame based approach and agent based approach. Later in 2006, [5] identified that current spoken dialogue system had been extended to multimodal dialogue system, which means that the dialogue systems can process two or more combined user input modes such as speech, pen, touch, manual gestures, gaze, and head and body movements, etc in a coordinated manner with multimedia system output. Bui meanwhile identified that the central module of the spoken dialogue system and multimodal system is the dialogue manager (DM). The function of the DM is to coordinate the activity of corresponding subcomponents in a dialogue system and its main goal is to maintain a representation of the current state of the ongoing dialogue.

[41] discussed each component in the spoken dialogue system and their functionalities. The involved components explained in [41] are as follows:

- *Speech recognition* which converts an input speech utterance consisting of a sequence of acoustic-phonetic parameters into a string of words.

- *Language understanding* which is the component analyzing a string of words with the aim of producing a meaning representation for the recognized utterance. The produced meaning representation can be used by the following dialogue management component.

- *Dialogue Management* is the control component of the interaction between the system and the user. It is also responsible for coordinating with other components of the system.

- *Communication with external system* is, for example, a database system, expert system, or other computer application.

- *Response generation* which is the specification of the message to be output by the system.

- *Speech output* is the component to apply text-to-speech synthesis or pre-recorded speech techniques to output the system's message.

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

Dialogue manager is the most important component in the (multimodal) dialogue system. The main functions of the DM include, coordinating with other components, identifying the intention of the user's intention and deciding what to respond to the user at what time steps. In [34], the main tasks of dialogue manager are identified as following:

- Updating the dialogue context on the basis of interpreted communication
- Providing context-dependent expectations for interpretation of observed signals as communicative behavior
- Interfacing with task/domain processing (e.g., database, planner, execution module, other back-end system), to coordinate dialogue and non-dialogue behaviors and reasoning

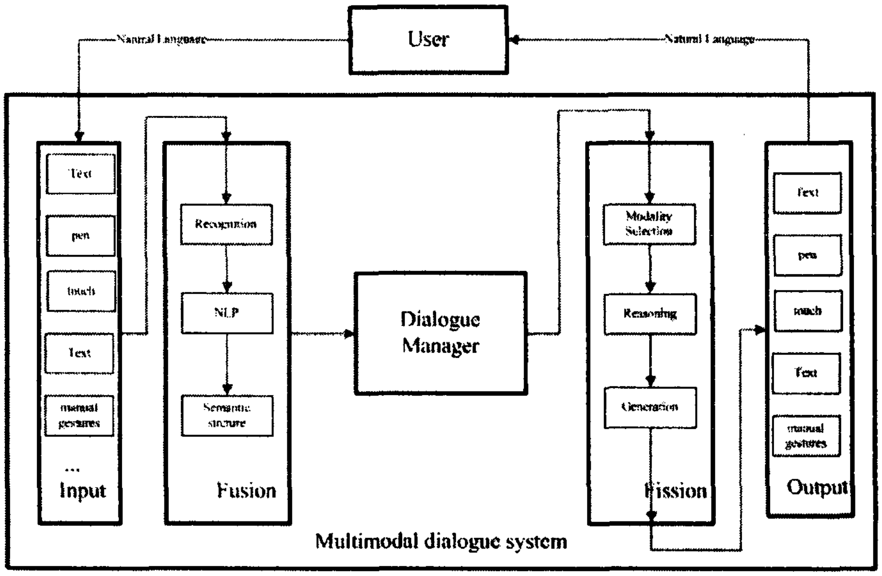

Figure 2.1: Overall Multimodal Dialogue System Structure

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

2.2 Dialogue Manager

Dialogue Manager is the core component of a dialogue management system. It simulates a task model in the specific domain. It processes semantic inputs from fusion and decides what the system should do next in response to the user in order to fulfil the user's goal.

Mctear in 2002 identified that DM may draw on a number of knowledge sources. And these knowledge sources are sometimes referred to collectively as dialogue model. The knowledge sources identified by Mctear [22] are the following:

- A *dialogue history* which records the dialogue proceeding so far in terms of the conditions that have been discussed and the information that has been mentioned. This representation provides a basis for conceptual coherence and for the solving the problem caused by anaphora and ellipsis.

- A *task record* is a representation of the information to be gathered in the dialogue. This record is usually in the format of a form, template or status graph. It is used to determine what information has not yet been acquired at the current stage. This record can also be used as a task memory [1] for cases where a user wishes to change the values of some parameters without needing to repeat the previous dialogue to require the other values that remain unchanged.

- A *world knowledge model* is the model containing general background information. This model can be used for the commonsense reasoning required by the system.

- A *domain model* is a model with specific domain information.

- A *generic model of conversational competence*. This model includes knowledge of the principles of conversational turn-taking and discourse obligations. For this model, it can be considered as the control level knowledge or the reward model to specify what is the appropriate action to be taken at the particular dialogue state.

- A *user model* is the model that may contain relatively stable information about the user, which may be relevant to the dialogue. With this model, it can provide the customized services by recorded user's age, gender, and preferences information. Also, the information such as user's goals, beliefs and intentions that changes over the course of the dialogue may be recorded.

After that the knowledge sources probably used in the DM are discussed, it comes to the problem of modeling based on the knowledge sources. Therefore, the distinction between dialogue modeling and dialogue management modeling [43] should be also explained here. The goal of dialogue modeling is to develop general theories of dialogues such as task oriented and to investigate the similarities between the course of the dialogues. Dialogue modeling is to provide dialogue management with theoretical support. While, the goal of dialogue management modeling is to combine dialogue model with task model under particular domain to design algorithms which support a machine's decision making in a dialogue, or it can be said that it takes the viewpoint of a dialogue system designer.

Dialogue manager executes based on the dialogue policy. Dialogue policy is "What the system should do next to respond to the users", which maps from a set of states in the state space to a set of actions. Usually, the actions of a dialogue manager can be divided into five types including: greeting, submitting, initiative, repeating and confirmation. And at each stage, the action taken by the system can receive different results or it can receive various rewards or costs. In some stage, such as the first round of the dialogue, the greeting should always be the most appropriate action taken by the system. While during the whole dialogue, the action of initiative, repeating and confirmation are always not clear. At each round, different actions correspond to different rewards. In the past efforts made on the dialogue management, various approaches have been proposed to resolve the dialogue policy generation problem.

Chapter 3

Related Work

In this chapter, the classification of the dialogue management approaches in the literature will be discussed in the first section. And in the following section, main approaches ranging from the finite state machine based to POMDP based approach for the dialogue management will be reviewed.

3.1 Dialogue Management Approaches Classification

Active investigations have been conducted in the past two decades towards dialogue management. The control strategy of a dialogue system may use finite states, frame slots, autonomous agents [22] or Bayesian networks and decision graphs approach [17]. Some dialogue strategies may be generated by the plan-based approach [27] which is based on the view that humans communicate to achieve goals, collaborative agent-based approach [2] which considers the dialogues as collaboration between two intelligent agents to achieve mutual understanding of the dialogue or theorem Proving approach [29]. Different dialogue management approaches have been classified into several categories by the researchers.

[43] categorized DM approaches into four including DITI (implicit dialogue model, implicit task model: Finite state-based models), DITE (implicit dialogue model, explicit task model: frame-based models), DETI (explicit dialogue model, implicit task model), DETE (explicit dialogue model, explicit task model). Another classification of three categories applies to dialogue grammars, plan-based approaches, and cooperative approaches [9][8]. However, all the different approaches are not mutually exclusive, or often used together [5]. For example, the frame based approach is usually used by combining with probabilistic method. Basically, there are main five types of dialogue management including finite state machine, frame base, Bayes network, Markov Decision Process based and POMDP based approach based upon the recent development of information state and probabilistic methods.

3.2 Information State Approaches

Finite state model based dialogue system is a basic system initiative dialogue management approach. The system directs the user with the all predetermined questions designed by the developers to complete some task. Finite state machine based approach models the dialogue flow and task model, each node stands for the system utterance and the edges correspond to the user's answers which determine all possible paths through the network. This approach is the most common and simple one. In this approach, the system collects one piece of information at a time and before submiting all the information to knowledge base or database, it will explicitly confirm with the user. Both task model and dialogue model are implicit and they are encoded by a dialogue designer. [19] discussed this approach and applied this approach in the Danish Dialogue Project. They used a basic finite state network to model the dialogue flow for an automatic book club service. More details about the theory of this approach are described in [9]. A dialogue model in the domain of the train ticket issuing system will be used in the following of this section to illustrate the various dialogue management approaches. In this domain, the ticket can be issued after both departure city and arrival city information obtained. The following Fig 3.1 illustrates the finite state machine based approach dialogue management under above mentioned domain.

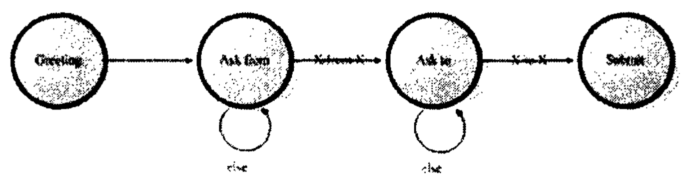

Figure 3.1: Finite State Machine based Approach Dialogue Management

[22] stated that the obvious advantage of the finite state model approach is simplicity and this approach is only suitable for the well-structured task. The questions to be asked and their sequences are predetermined. In the whole dialogue session, the agent guides the user and constrains the format of the user's answers. After each turn in the dialogue, the agent will explicitly paraphrase with the user what have been said just now. As there are so many restrictions on the dialogue, the agent does not require advanced technology applied such as natural language processing. The advantage of the finite state based approach meanwhile reflects the disadvantages including: it can only apply to the simple domain, lacks of flexibility in the dialogue. During the dialogue, the user neither influences the dialogue nor brings in new dialogue topic. When the more uncertainties brought by the users or environment, the system can easily crash because of the inappropriate dialogue policy generated by the domain's expert and restricted preset script.

In finite state based approach, the dialogue system is agent directed and only collects one piece of information at each turn based on its current dialogue state. When the user introduces more information than the system requires at each dialogue state, it comes the problem. Finite state machine based approach neither realizes the multiple slots filling nor deals with the redundant information brought by the user. As an extension of finite state-based model, frame-based model is developed to overcome the lack of flexibility of the finite state machine based approach.

Regarding frame-based approach, it's like a task of slot-filling, which a slot is a predetermined set of information that should be gathered by the agent and the dialogue conducted by the unfilled slot. Frame based approach allows some degree of mixed-initiative and multiple slot fillings, which resolves the problem of the finite state machine based approach. However, the dialogue model is still encoded by a dialogue designer based on their experiences and understandings. The frame based approach is illustrated in Fig 3.2:

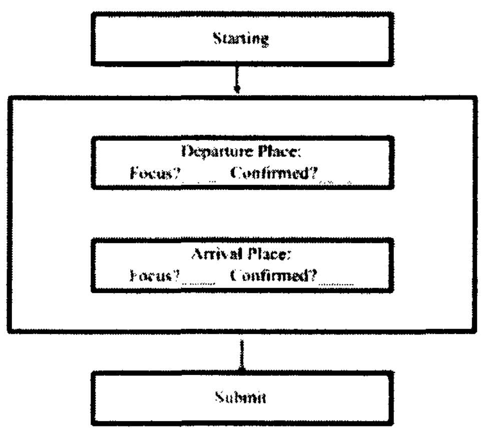

Figure 3.2: Frame-based Approach Dialogue Management

Ward and Pellom in [36] used the similar mechanism in their communicator system, in which the next action of the agent is generated based on the current context rather than preset script. Dahlback and Jonsson in [11] used information specification forms under the domain of bus timetable information system. A more flexible frame based approach was proposed by Goddeau et al. [15] named E-form which has been applied in a spoken language interface to a classified advertisements for used car database.

Bui also summarized other variants of frame-based models which allow to deal with more complex dialogues. These variants include: schemas, agenda are used in the Carnegie Mellon Communicator system to model more complex tasks [10], [26], [42], [3], task structure graphs which provide a semantic structure and are used to determine the behavior of the dialogue control as well as the language understanding module [40], type hierarchies are used to model the domain of a dialogue and as a basic for clarification questions [12], blackboard is used to manage contextual information relevant to dialogue manager such as history board, control board, presentation board, etc [24].

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

3.3 Probabilistic Approaches

Recently, several groups of researchers have been working on probabilistic approaches to improve the performance of dialogue management. They can be considered as the extension of information state approaches [34].

Wai et al. [35] proposed to use of Belief networks (BN) for mixed-initiative dialogue modeling. They applied their approach into the CU FOREX system which is a bilingual hotline for real time foreign exchange inquiries. The author adopted Belief networks in mixed-initiative dialog modeling involving the following two processes: inferring the informational goal of a user's query and verifying the input query against domain-specific constraints. In the process of goal identification, a BN is trained for each domain-specific informational goal and then it is used to make a binary decision based on the concepts present in the input query. With the decisions across all BNs combined, the output goal can be identified regarding the input query. Followed by the backward inference process, the validity of the input query will be verified. The system responses can be generated based on the result of the spurious and missing concepts detection process. In 2003, Wai et al. migrated their dialog model from the simple foreign exchange domain to air travel information service domain. In this work, they described the scalability and portability of a Belief Network based mixed initiative dialog model across application domains.

Keizer et al. [18] stated the problem that utterance features were more informative and this phenomena increased the difficulty in classifying the utterance to speech act. They also claimed that a dialogue system needs a better user model and the ability to understand the user's intentions depends on the user model. They proposed to use Bayesian networks as a user modeling method for the dialogue act recognition of a dialogue system for Dutch dialogues. In the process to improve the user model to decrease uncertainties brought by the user utterance.

Paek and Horvitz [23] proposed using Decision Networks as the dialogue model to manage a hidden subdialog. Paek and Horvitz stated that the problem that when the dialogue system attempts to solicit information from the user, it may have to engage in a hidden sub dialog or error handling in a particular state. They considered that hidden subdialogs generally centers on illocutionary repairs including asking for repeating or conforming, etc. It is described that there are three advantages by applying the decision network in dialog management: first the propagation of uncertainties over time to assist recognition, second the ability to leverage key contextual dependencies, such as the acoustic environment, and the consideration of the stakes involved in taking real-world actions. Williams et al. in [37] stated that this approach selects the action only based on the immediate maximum expected utility and in this scenario this proposal can be treated as a POMDP that greedily selects the actions.

Another important groups of researchers have delved their efforts into probabilistic techniques such as (fully observable) Markov Decision Process (MDP) or a Partially Observable Markov Decision Process (POMDP) as the dialogue model to resolve the action outcome and observation uncertainties existed in the human-computer interaction process. [21] and [28] all cast the dialogue management problem as the MDP problem with the assumption that a good dialogue strategy is minimizing an objective function that reflects the costs of all the important dialogue dimensions. Levin et al. stated that allowing a user to change the course of dialogue or to change request during dialogue in a mixed-initiative system could result in a branching factor and make the tree prohibitively large. Therefore, they adopt the Markov Decision process approach. The operation of the dialogue manager based on the Markov Decision Process described by Levin et al. is as followed:

Initialization: start from initial state

Iterate until done (final state is reached)

Next Action: Choose and perform next action

Get new input

Next State: Update state with new input

In 2002, Singh et al. proposed to apply Reinforcement Learning to dialogue policy design. They applied the reinforcement learning algorithm to design optimal strategy automatically. The methodology was described as following [28]: choose an appropriate reward measure for dialogue, an appropriate representation for dialogue states, and design a dialogue policy that maps each state to asset of reasonable actions. Build an initial state-base training system that creates an exploratory data set. Despite being exploratory, this system should still provide the desired basic functionality. Using theses training dialogues, an empirical MDP model is built on the state space. The transitions of this MDP will be modeling of the user population's reactions and rewards for the various system actions. Optimal dialogue policy computation is according to this MDP and then the system is re-implemented using the learned dialogue policy. This process can be illustrated in Fig3.3 :

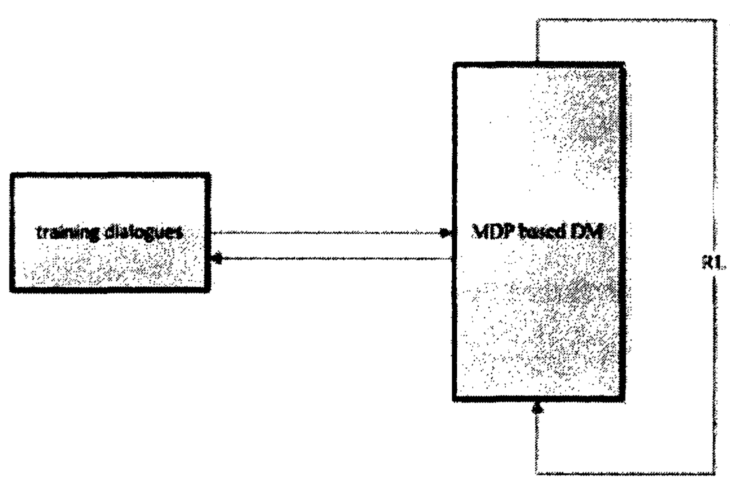

Figure 3.3: MDP with Reinforcement Learning Approach to DM

Roy et al. was the first group to treat dialogue management as a problem of partially observable Markov decision process [25]. They noticed that the MDP approach cannot handle noise and ambiguity in speech utterances. They used POMDP models to generate dialogue strategy and used, rather than estimated system state, belief state to represent user intention. They conducted experiments and claimed that the POMDP-based dialogue system made fewer mistakes than MDP-based dialogue system. With increased errors in automated speech recognition in real-life situations, the advantage of uncertainty handling is obvious.

Zhang et al. made an extension by adding "hidden" system states and using Bayesian networks to combine observations from a variety of sources [45]. Williams et al. further proposed a factored architecture to add a component from the perspective of user and to allow dialogue designers defining more appropriate reward measures [37]. They also improved automated planning with combining confidence scores [38]. In this work, the composite observation which contains discrete elements representing dialogue acts and continues components representing confidence scores is used. The improved approach with the continues confidence score performs better than the traditional approach. Also, this method can be used to improve the handcrafted dialogue manager. To solve the scale up problem of POMDP based approach, composite summary point-based value iteration algorithm was proposed by [39]. Under this method, for each slot there is a local POMDP solution will be created and each local solution gives an action. A heuristic choose will be used to decide what action to be taken next. The scale-up problem of POMDP-based approach was also addressed with a hidden information state model [44]. Young et al. in their work proposed a hidden information state model which can scale. This model can be used in practical system development. The prototype system which is in the tourist information domain was developed and the results demonstrated that this approach can build a robust spoken dialogue system. The application of POMDP-based approach in effective dialogue management was also studied by Bui in [4]. In Bui's work, the state of POMDP model is extended with a user effective factor. A single-slot route navigation dialogue problem was studied and used to prove the concept. With this approach, the performance of the dialogue manager can be improved given the user's affect state.

Chapter 4

An Analysis With History Information Space

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

4.1 Overview of History Information Space

LaValle identified that in the human-computer interaction problem or problem relating to applications interaction with physical world, the sensor obtains the information regarding the state. Usually, the information obtained from the sensor is limited and state is estimated based on the limited information. When the estimates are sufficiently reliable, the action will be taken with assumption that there is no uncertainty existing. In this case, the planning problem expressed in terms of an history information space can avoid state estimation, which resolves the problems of sensor uncertainty. LaValle [20] defined the history information space as the history including sensor observation history, actions have been applied and the initial condition. Sensor is designed to sense the state and it consists two parts including an observation space which is the set of possible readings and a sensor mapping which characterizes the readings that can be expected at the current state or given other information. Three important sensors including state sensor mapping, state-nature sensor mapping and history-based sensor mapping were defined by LaValle.

The formal definition of history information space is as following [20]: The set of all observation histories is denoted as $\widetilde{Y}_k$ and is obtained by a Cartesian product of k copies of the observation space:

$$\widetilde{Y}_k = \mathrm{Y} \times \mathrm{Y} \times \mathrm{Y} \ldots \times \mathrm{Y}$$

The set of all action histories is $\widetilde{U}_{k-1}$, the Cartesian product of k-1 copies of the action space U. Planning under information space is based on the information state which is always known. I_0 denotes the initial condition space, the above mentioned known state which means the initial state x_1 is given, then $I_0 \in$ X. At the stage k or time step k, the history I-space at stage k is expressed by the following:

$$I_k = I_0 \times \tilde{U}_{k-1} \times \tilde{Y}_k$$

With the definition of the observation history and action histories, the definition of the history information space is the union of each information stage I_k over all k $\in \{0\} \cup \mathrm{N}$ as the following:

$$I_{hist} = I_0 \cup I_1 \cup I_2 \cup \cdots \cup I_k \tag{4.1}$$

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

LaValle also gives the definition of discrete information space planning which is illustrated in Fig 4.1:

4.2 Dialogue Management Approaches Analyzed Upon History Information Space

According to the theory of information space [20], the only information available to a decision process at stage k of a dialogue is the history of all observations $\widetilde{Y}_k$ at that stage and the history of all actions $\widetilde{U}_{k-1}$ that have been taken before that stage. Let Y, U denote the observation space and the action space respectively. Given an initial condition η_0, $\widetilde{Y}_k$ and $\widetilde{U}_{k-1}$ are two Cartesian products of observation and action spaces respectively at their

1. A nonempty *state space* X that is either finite or countably infinite.
2. A nonempty, finite *action space* U. It is assumed that U contains a special *termination action*.
3. A finite *nature action space* $\Theta(x,u)$ for each $x \in X$ and $u \in U$.
4. A *state transition function* f that produces a state, $f(x,u,\theta)$, for every $x \in X$, $u \in U$, and $\theta \in \Theta(x,u)$.
5. A finite or countably infinite *observation space* Y.
6. A finite *nature sensing action space* $\Psi(x)$ for each $x \in X$.
7. A *sensor mapping* h which produces an observation, $y = h(x,\psi)$, for each $x \in X$ and $\psi \in \Psi(x)$. This definition assumes a state-nature sensor mappings. A state sensor mapping or history-based sensor mapping could alternatively be used.
8. A set of *stages*, each denoted by k, which begins at $k = 1$ and continues indefinitely.
9. An *initial condition* η_0, which is an element of an *initial condition space*, $\mathcal{I}_0$.
10. A *history I-space* $\mathcal{I}_{hist}$ which is the union of $\mathcal{I}_0$ and $\mathcal{I}_k = \mathcal{I}_0 \times \tilde{U}_{k-1} \times \tilde{Y}_k$ for every stage $k \in \mathbb{N}$.
11. Let L denote a stage-additive cost functional, which may be applied to any pair $(\tilde{x}_{K+1}, \tilde{u}_K)$ of state and action histories to yield

$$L(\tilde{x}_{K+1}, \tilde{u}_K) = \sum_{k=1}^{K} l(x_k, u_k) + l_F(x_{K+1}).$$

If the termination action u_T is applied at some stage k, then for all $i \geq k$, $u_i = u_T$, $x_i = x_k$, and $l(x_i, u_T) = 0$. Either a feasible or optimal planning problem can be defined; however, the plan here is specified as $\pi : \mathcal{I} \rightarrow U$.

Figure 4.1: Defining Discrete State Planning Problem in History Information Space

corresponding stages.

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

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

If η_0 belongs to an initial condition space I_0, a history information space I_{hist} is formed as the union of I_0 and $I_k = I_0 \times \tilde{U}_{k-1} \times \tilde{Y}_k$ for up to the kth stage.

$$I_{hist} = I_0 \cup I_1 \cup I_2 \cup \cdots \cup I_k \tag{4.2}$$

An information-feedback plan $\pi = (\pi_1, \pi_2, \cdots)$ then maps I_{hist} into a sequence of actions $\mu_1, \mu_2, \cdots \in U$.

$$\pi : \quad I_{hist} \rightarrow U \tag{4.3}$$

An optimal plan π^* maximizes a given stage-additive cost function.

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

Among the major approaches of dialogue management, the FSM-based approach uses a predetermined sequence of system actions. The user neither has any initiatives nor influences the control during the whole session of dialogue. The action or policy generated by the system takes no feedback from the user either. If η_k is the state of information space at the kth stage, the information mapping function f becomes $f_k(\eta_k) = k$. Correspondingly, I_{hist} is simplified into a derived information space in which only the stage number k is maintained at every stage. Planning as formulated in Eq. 4.3 now follows the policy below, where $\mathbb{N}$ is the set of stages.

$$\pi_{stg}: \ \mathbb{N} \rightarrow U \tag{4.4}$$

Eq. 4.3 helps the FSM-based approach to dramatically reduce the size of the history information space, and produce a working plan for dialogue management. However, the abandon of all information in I_{hist} but the stage number makes this approach the least flexible in human-robot interaction.

As an improvement, the frame-based approach takes the current context into consideration when deciding the next action. In this approach, information contained in the slots of the frame (or form) is used to estimate the current state of the dialogue system, and the estimated state is then used to generate responses to the user. At this time, function f becomes an information mapping $f: I_{hist} \rightarrow X$, where X is the set of estimated system states. Consequently, the following equation decides the planning process.

$$\pi_{est}: \ X \rightarrow U \tag{4.5}$$

In such a way, the frame-based approach makes use of the historical value of slots in the frame, but drops all other types of information, including the action history. As a result, the plan cannot distinguish different histories, and decision will only be made based on the current slot values which provide no hint about the history of change in values. In addition, it is difficult to consider the long term influence when action are selected to maximize immediate reward.

Similarly, the approaches based upon both Bayesian network and MDP rely on estimated system state for their planning processes, and therefore they share the same equation (Eq. 4.5) with the frame-based approaches. In the two cases, the simplification of I_{hist} is accomplished by keeping only the observation history when estimating the current state of the system. All the information of action history is ignored. Between the two, the Bayesian network-based approach uses a trained Bayesian network to decide actions according to the estimated system state, and the MDP-based approach conducts computations with an iteration algorithm for action selection.

4.3 Conclusion

In this chapter, main dialogue management approaches are reviewed again and analyzed upon history information space. Dialogue management approaches can be considered as a planning problem under history information space. Under history information space, each approach equals information mapping function in the history information space which compresses the information space to a derived space. Different dialogue policies are generated based on the derived information space. For each generated dialogue policy, it characterizes appropriate machine action should be taken at the current stage or given information. For the FSM-based approach, it completely destroys the history information space and from frame-based approach to the MDP-based approach, they drop different information at various level to some degree to perform dialogue management. Except for the current

POMDP-based approach, MDP-based approach achieves best performances. However, all those approaches analyzed in this chapter maintain the concept of the machine state and performance of the dialogue depends on the quality of the machine state estimation.

Chapter 5

The Proposed Method

Dialogue management is fundamentally a problem of planning under the influence of uncertainty. This chapter first uses the theory of information space to examine the POMDP-based approach of dialogue management, and then proposes a new approach for better recognition of user intention. The advantages of the new approach will be demonstrated with experiments in the next chapter.

5.1 POMDP Models

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

5.1.1 Flat POMDP Model

In a POMDP system, the state of the system is not observable and therefore unknown to the decision process. Action selection depends on the decision made over belief state, denoted by b. Formally, a POMDP is defined as a tuple $\{S, A_m, T, O, Z, R\}$, where S is a set of states, A_m is a set of actions the system may take, T is the transition model that defines transition probability, O is a set of observations from user's actions, Z is the observation model that defines the observation probability, and R defines the immediate expected real-valued reward $r(s, a_m)$. And also b is the agent's belief state and π is the agent's policy to select action.

With the assumption that the state and O are both discrete and based on the above notation and definition, the operation process of POMDP can be described as following: POMDP system carries out two tasks. The first task is to compute or update belief state, and the second is to find an optimal policy [4]. With the latest belief state and the offline computed optimal policy, the agent can perform appropriate action checking to select action to take.

For the first task, the belief state is updated at each time step based upon the Bayes filter algorithm. Bayes filter algorithm is under the Markov assumption. The Markov decision process makes assumption that the action of nature only depend on the current state and action as opposed to the state or action histories. In [33], the Markov Assumption and Bayes Filters framework were describes as following. Markov assumption is with underlying assumption that the world is static, the noise is independent and perfect model no approximation error. The Markov assumption allows the recursive Bayesian updating to be used to efficiently combine evidence. The Markov Assumption illustrated in Dynamic Bayes network is shown in Fig 5.1.

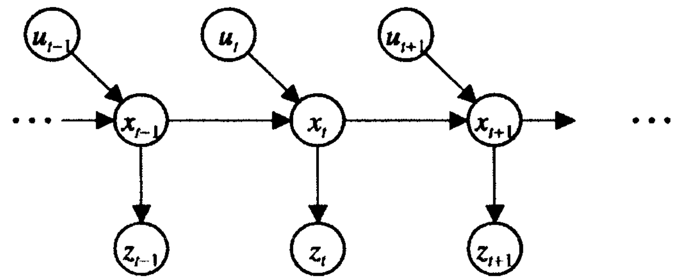

Figure 5.1: Markov Assumption illustrated in Dynamic Beyes Network

The expected observation probability depends on local information. Here the local information means that distribution depends only on information obtained at the current stage. And the posterior probability over state depends on the previous state and newly taken action. In general, it's described with the following equations:

$$p(z_t|x_{0:t}, z_{1:t}, u_{1:t}) = P(z_t|x_t)$$
$$P(x_t|x_{1:t-1}, u_{1:t}, z_{1:t}) = P(x_t|x_{t-1}, u_t)$$

Based on the above Markov assumption, Bayes filter is the probabilistic method to estimate state in dynamic environment. The estimation of state computation process is shown in Fig 5.2.

Thus, the computation of belief state uses the following equation, where α is the normalizing constant, $P(o_{t+1}|s_{t+1}, a_t)$ is the observation model or named sensor model and $P(s_{t+1}|s_t, a_t)$ is the action model or named transition model.

$$b_{t+1}(s_{t+1}) = \alpha P(o_{t+1}|s_{t+1}, a_t) \sum P(s_{t+1}|s_t, a_t) b_t(s_t) \tag{5.1}$$

$$
\begin{aligned}
Bel(s_t) &= P(s_t \mid a_1, o_1 \ldots, a_t, o_t) \\
&= \alpha\ P(o_t \mid s_t, a_1, o_1, \ldots, a_t)\ P(s_t \mid a_1, o_1, \ldots, a_t) \\
&= \alpha P(o_t \mid s_t)\ P(s_t \mid u_1, o_1, \ldots, u_t) \\
&= \alpha\ P(o_t \mid s_t) \int P(s_t \mid a_1, o_1, \ldots, a_t, s_{t-1}) P(s_{t-1} \mid a_1, o_1, \ldots, a_t)\ ds_{t-1} \\
&= \alpha P(o_t \mid s_t) \int P(s_t \mid a_t, s_{t-1})\ P(s_{t-1} \mid a_1, o_1, \ldots, a_t)\ ds_{t-1} \\
&= \alpha P(o_t \mid s_t) \int P(s_t \mid a_t, s_{t-1}) P(s_{t-1} \mid a_1, o_1, \ldots, o_{t-1})\ ds_{t-1} \\
&= \alpha P(o_t \mid s_t) \int P(s_t \mid a_t, s_{t-1})\ Bel(s_{t-1})\ ds_{t-1}
\end{aligned}
$$

Figure 5.2: Estimation of State using Bayes Filter

For the current belief state, Eq. 5.1 constitutes the flat POMDP model that selects an optimal policy as the maximum of all the expected value function $V^{\pi}(b)$ with a discounted future reward starting from b for a policy π.

$$\pi^* = argmax_{\pi} E[V^{\pi}(b)] \tag{5.2}$$

5.1.2 Factored POMDP Model

[37] cast a spoken dialogue system as a factored POMDP which extends the flat POMDP by decomposing the state into three components. The factored POMDP model extends the unobserved state to include the user action model, which is the user's most recent action and relevant dialogue history information from conversation. [37] illustrates the factored POMDP model as Fig :5.3:

Under the factored POMDP model, it assumes that user's goal depends only on the

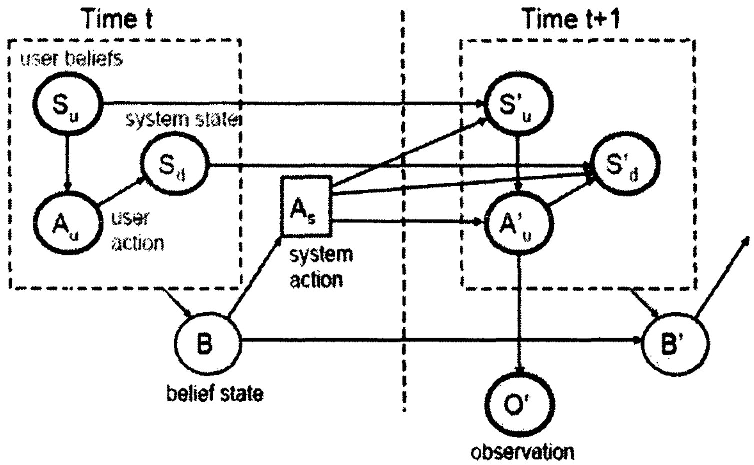

Figure 5.3: Factored POMDP

previous goal and the agent's action at each time step, the user's action depends on her or his current goal and preceding machine action, and the current dialogue state depends on the previous history along with the latest user and agent action. With this assumption, the extended unobserved state helps to revise Eq. 5.1 into the following belief update equation with more appropriate reward.

$$b'(s'_u, s'_d, a'_u) = \alpha P(o'|a'_u) P(a'_u|s'_u, a_m) \sum P(s'_u|s_u, a_m) \sum P(s'_d|a'_u, s_d, a_m) \sum b(s_u, s_d, a_u) \quad (5.3)$$

5.2 Shortcoming with the Current Models

The POMDP-based approach avoids the need to estimate system state by using a set of probability distribution over belief state in the planning process. Together with the action at the kth stage, previous belief state, the system uses new observations to update the belief state and plans for action at the next stage. In the process, the state of the system and the user is hidden in the information space. As defined in Eq. 5.1 for the flat modal and Eq. 5.3 for the factored modal, history information state is mapped to a probability distribution over the unknown system state. As this conversion is based upon on the Bayes filter theory, which in turn is under the Markov assumption, the POMDP-based approach plans for actions with only the current belief state, which is clearly illustrated in the b elements in both Eq. 5.1 and Eq. 5.3.

In POMDP models, actions A_m at the previous stage lead to observations probability Z at the kth stage, which corresponds to the Cartesian product of $\tilde{U}_{k-1} \times \tilde{Y}_k$ in I_k. It is a simplification of I_{hist} into I_k, resulting a complete loss of history information, including changes in belief states, series of observations, and sequences of actions. The simplified version of Eq. 4.3 uses the following formula.

$$\pi' : \; I_k \rightarrow U \tag{5.4}$$

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

In a another perspective, the POMDP-based dialogue management approach only models the user's goal or it can be considered as a user modeling rather than a task modeling or machine state modeling. Although, when dealing with the observation uncertainties and action uncertainties, the POMDP-based approach outperforms than other approaches. This advantage is even more obvious when the error rate of the input is high. The POMDP-based approach tries to listen correctly at its best. While, what if the user's goal it's trying to listen is not correct itself at the beginning? The task will finally end up with the failure although DM listens correctly. Usually the dialogue systems make a assumption that the user can always answer the questions from the agent. However, in the real life condition, the user is always lack of domain knowledge and provides unreasonable information to the agent. This situation will be worst when the user can not actually understand the question generated by the agent. If the dialogue management approach only models the user without its own domain knowledge level inference, the task can not achieved. In the process of the human computer interaction, if the computer can appropriately influence the user and guide the user, the task is more probably to be achieved.

5.3 A New Approach with Modification

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

$$\pi_{new}: \quad I'_{k-1} \cup I_k \rightarrow U \tag{5.5}$$

In the new approach, both I_k and I'_{k-1} are still in the form of belief state, and state updating still uses the existing POMDP models in described in Section 5.1.1. The addition of I'_{k-1} in the modified approach, however, introduces an important element to dialogue management, i.e., the history of belief state or the dynamics of belief state. Although the historical information of observations and actions is not maintained explicitly in I'_{k-1}, the union I_k and I'_{k-1} in Eq. 5.5 diminishes the negative effect of Markov assumption and allows POMDP-based dialogue management to plan for actions with not only the current belief state but also the updated history before reaching the current state.

The uncertainties that the original POMDP-based approaches fail to handle mainly arise from situations in which the user lacks knowledge in the domain or the user's goal cannot be fulfilled due to real-life constraints. In addition, dependency of factors in belief state also causes uncertainty. The original POMDP-based approaches is only able to resolve those uncertainties that are brought in by noise from observations, e.g., misinterpretation of words, and actions, e.g., misunderstanding of meaning. The dialogue system tries to "listen correctly" and to response appropriately to the user based on its state of belief. By interrupting the planning process of POMDP-based dialogue management, a new component can be added to introduce a knowledge base with new rules and a database with practical constraints.

Shown in Fig. 5.4 is the architecture for the modified POMDP-based dialogue man-

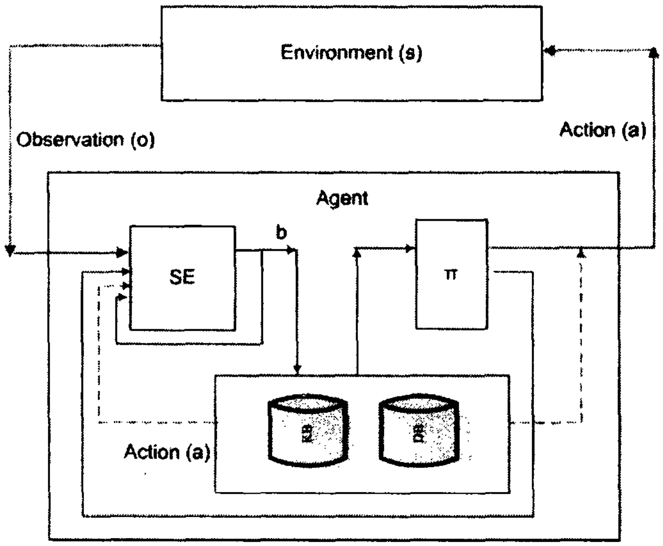

Figure 5.4: Architecture of the Proposed Approach

agement, in which the additional component interrupts the direct flow from b to π. As a realization of the new planning strategy (Eq. 5.5), action a alters the original action when there is an unexpected change from I'_{k-1} to I_k, or more accurately from the previous belief state to the current state. The added component also skips the original planner π and makes direct contact with the user. The algorithm for the new approach is shown with a flow chart in Fig. 5.5. After an initial greeting, the system always updates the belief state with previous belief state, the current action, and the latest observation from the user. At each stage of a dialogue, the new approach uses the domain knowledge and constraint database to help validating the change of belief state. A failed validation results a roll-back of belief state to the previous stage. Meanwhile, it triggers a explanation to the user and a question requesting further information. This planning process is able to guide the user reaching a feasible goal that satisfies the need without causing conflicts. The Fig 5.6 is the pseudo code of the proposed new approach.

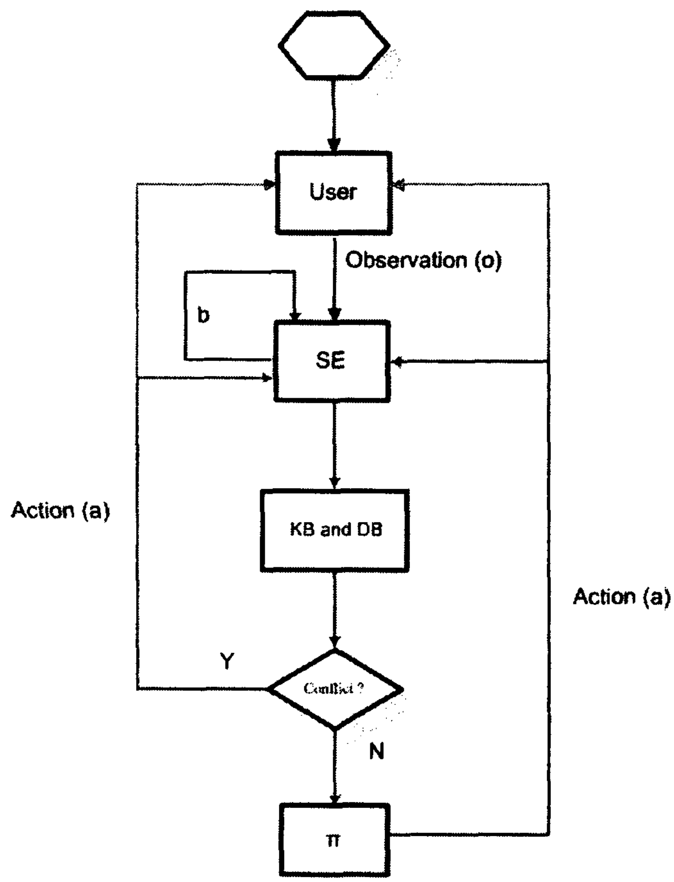

Figure 5.5: Flow Chart of Proposed Approach

New_proposed_method (*Bel(s), o, a*):

bp= *Bel(s)*

For all *s* do

    Bel'(s) $= \alpha P(o_{t+1} \mid s_{t+1}, a_t) \sum_{s_t \in S} P(s_{t+1} \mid s_t, a_t) b_t(s_t)$

For all s

    If *Bel'(s)* < 0.05

        Delete *Bel'(s)*

    else

        DomainConstrain_validation (*Bel'(s)*)

If all pass validation

    Machinenext_action=out.policy(*Bel'(s)*)

    return Machinenext_action

else

    Machinenext_action= *a* with the hint

    Bel'(s)= bp

    return Machinenext_action

Figure 5.6: Pseudo Code of the Modified Approach

In the pseudo code of Fig 5.6, the *Bel(s)*, o and α are the inputs of the proposed method. *Bel(s)* is the previous belief state of the last stage, o is the latest observation and α is the last action taken by the machine. By recording the previous belief state, the belief state will be updated based on the POMDP theory and for all the belief states with possibility less than 0.05. For all the rest of the belief states, the domain constraint validation process function *DomainConstrain_validation()* will be invoked to check the conflicts of the belief states. The failure validation will result in the action to require further information with hints and the roll back of the belief states, otherwise the action produced by the original POMDP solution policy *out.policy(Bel'(s))* will be taken.

Chapter 6

Experiments And Discussions

In this chapter, the domain background applied in our case study will be introduced first and the implementation platform, utilized tools and corresponding details will be explained after. At the end of this chapter, the results under three scenarios and results analysis will be given.

6.1 Applied Domain Background

Experiments are conducted based upon a simulated situation in which an agent provides assistance at a computer shop to a human user for the purchase of a computer. In this domain, the configuration of a computer is supposedly determined only by three factors — number of CPU cores (single or dual), types of netbook or laptop computer, and its price tag at $600 or $1,200. The user is not expected to have the domain knowledge, and therefore may make unreasonable requests. The user, however, is expected to be rational who is ready to change the goal after an explanation by the agent. For any computer, its price depends on the core number and computer type. It is also assumed that a netbook cannot have a dual core, and a laptop with dual core cannot cost only $600.

6.2 Implementation

Experiments are programmed in JAVA language on a laptop running Linux Ubuntu 10.04 version, and the POMDP-based approach of dialogue management uses Eq. 5.3 for its implementation. The testing system uses the structure in Fig. 5.4, and the new approach follows the flow chart in Fig. 5.5.

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

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

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

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

The reward model are specified in this R: <action> : <start-state> : <end-state> : <observation> %f format. For any of the entries appeared in the above, an asterick * for either <state>, <action>, <observation> indicates a wildcard which means this item will be expanded to all existing entities.

```
discount: %f
values: [ reward, cost ]
states: [ %d, <list of states> ]
actions: [ %d, <list of actions> ]
observations: [ %d, <list of observations> ]
```

Figure 6.1: Tony Cassandra POMDP Specification File Starting Line

Dialogue specification file parser is using a more concise format of POMDP specification which uses regular expression. The parser can read POMDP specification file or factored POMDP specification file in a more concise format using regular expression and convert the concise specification file to Canonical POMDP file which is in the Tony Cassandra's format. This Dialogue specification parser really save lots of efforts in specifying the domain problem especially when it comes to a complicated domain which has more than thousands states. Dialogue specification parser can convert both *dlgpomdp* format and fpomdp format.

In the experiments, dlgpomdp formate is adopted. Under this format, a number of fields are consisted and each filed starts with the name of the field on a new line followed by a colon. A list of all fileds is below the filed name. For each field, every item between the colon is specified by the regular expressions. In order to capture matches, parentheses in a pattern can be used. Also, referring to the captures can be made by using tokens $1, $2 etc in the following regular expression at the same line. The same as POMDP specification file in Tony Cassandra's format, the dlgpomdp file should also start with the 5 lines which specify user goals, user actions, dialogue states, system actions and discount values. The order of this 5 line can also be in any sequences. Below the staring lines, there are 7 fields needing to be specified as following:

1. Legal state field specifies the legal state of the agent. The legal state is specified in the following format: *user goal : user action : dialogue state*. Any state matches one of the specification is the legal state.

2. Start state field specifies start state in the same above mentioned legal state format. All start states have an equal possibility of belief states.

3. User goal model field specifies that the possibility that the goal is change to another under particular system action. The format is user goal : *user goal : system action : user goal' possibility value* . Any entity of user goal, system action and new user goal matches the regular expression, then the value of this user goal function is the specified value at the end of the line. Any value doesn't match the function, the possibility is the 0.0.

4. User action model field is in the format of *dialogue state : system action : user goal : user action* and specifies the possibility of the next user action when given the current dialogue sate, system action and user's goal. The same with the user goal model, any matched items have the specified possibility of the function and otherwise the possibility is 0.0.

5. Dialogue history model field specifies the possibility of dialogue changing to another under corresponding system action, the current user goal and the user action. The transition portability of dialogue state is specified in the format of *dialogue state : system action : user goal : user action : dialogue state possibility value*

6. Condition for transition to end state field specifies under what condition the dialogue will transit to the end.

7. Reward model field is in the format of *user goal : user action : dialogue state : system action reward value.* It specifies what reward will be gained at the current dialogue state with the user's goal and action. If the entities do not match any R field, then the value of the reward function is 0 for that entities.

8. Observation model field specifies the possibility of the observation with corresponding user action.

For the simulated situation computer purchase, the dolpomdp specification format is designed based on the experiences and domain knowledge. There are 8 user goal specified and user may take 27 kinds of actions. The system can perform 13 types of actions. The number of the dialogue state is 9 including the special ending state, f. The discount value is 0.95 in this experiment. With all the specification, there are total 1944 states in the complete state space. The states which pass the legal state parser contained in the Dlgpomdp specification file will produce the valid state for the agent to make decision on. The starting 5 lines of variables definition are specified in Fig 6.2 and different model specification examples are illustrated in the Fig 6.3.

The POMDP solver adopted in this experiment is ZMDP [30] solver. ZMDP is a software package which implements several heuristic search algorithms for POMDPs and MDPs developed by Trey Smith at the Carnegie Mellon University. ZMDP POMDP solver can work under both Linux and Mac operation system. To solve the POMDP problem in our experiments, heuristic search value interaction algorithm (HSVI) [31][32] are used. HSVI is a point-based algorithm that maintains both upper and lower bounds on the optimal value function, allowing it to use effective heuristics for action and observation selection, and

```
user-goals:      core1typeLprice600, core1typeLprice1200, core1typeNprice600,
                 core1typeNprice1200,core2typeLprice600, core2typeLprice1200,
                 core2typeNprice600, core2typeNprice1200

user-actions:    core1typeLprice600, core1typeLprice1200, core1typeNprice600,
                 core1typeNprice1200, core2typeLprice600, core2typeLprice1200,
                 core2typeNprice600, core2typeNprice1200, core1typeL, core1typeN,
                 core1price600, core1price1200, core2typeL, core2typeN,
                 core2price600, core2price1200, typeLprice600,
                 typeLprice1200, typeNprice600, typeNprice1200, core1,
                 core2, typeL, typeN, price600, price1200, null

dialogue-states: nnn, nns, nsn, nss, snn, sns, ssn, sss, f
                  # f is finished: system said bye or gave correct answer

system-actions:  core1typeLprice600, core1typeLprice1200, core1typeNprice600,
                 core1typeNprice1200, core2typeLprice600, core2typeLprice1200,
                 core2typeNprice600, core2typeNprice1200,
                 askcore, asktype, askprice, hello, bye

discount: 0.95
```

Figure 6.2: Variables Definition of Dlgpomdp Specification File

```
# Legal states
legal-states:
(core[0-9]type[A-Z]price[0-9]*) : $1 : sss
(core[0-9]type[A-Z])price[0-9]* : $1$ : ss.
...
# Start states
start:
core[0-9]type[A-Z]price[0-9]* : null : nnn

# User goal model
SU: (core[0-9]type[A-Z]price[0-9]*) : ^[A-Za-z0-9]+$ : $1 1.0
...
# User action model
# dialogue finished
AU: f : ^[A-Za-z0-9]+$ : ^[A-Za-z0-9]+$ : null 1.0
...
# 'type''core' correcnt 'price' wrong
AU: .* : (core[0-9]type[A-Z])price[0-9]* : $1price([0-9]*) : ^price$2 0.7
AU: .* : (core[0-9]type[A-Z])price[0-9]* : $1price([0-9]*) : ^$1price$2 0.3
AU: .* : (core[0-9]type[A-Z])price[0-9]* : $1price([0-9]*) : ^[A-Za-z0-9]+$ 0.0
...
# Dialogue history model
SD: f : ^[A-Za-z0-9]+$ : ^[A-Za-z0-9]+$ : ^[A-Za-z0-9]+$ : f 1.0
...
# Condition for transition to end state
transition-to-end:
^[A-Za-z0-9]+$ : ^[A-Za-z0-9]+$ : .* : bye
# Reward model
R: core([0-9])type([A-Z])price([0-9]*) : ^[A-Za-z0-9]+$ : sss : core$1type$2price$3 10
R: core([0-9])type([A-Z])price([0-9]*) : ^[A-Za-z0-9]+$ : f :  core$1type$2price$3 -1
...
# Observation model
O: (^[A-Za-z0-9]+$) : $1 0.8
O: (^[A-Za-z0-9]+$) : ^[A-Za-z0-9]+$ 0.2/26
```

Figure 6.3: Examples of Different Model Specifications in Dlgpomdp Specification File

to provide probably small regret from the policy it generates [32]. A more details of the HSVI algorithms can be found in [32]. By receiving the POMDP specification file in Tony Cassandra's format, the ZMDP solver produces the *out.policy* file which specifies each hyperplane along with corresponding approximate optimal solution. In POMDP policy file, a set of "lower bound planes" which consists of an alpha vector and a corresponding action are presented. With a current belief b, the lower bound on the expected long-term reward starting from b and which action can achieve the expected lower bound can be known. In this experiment, the ZMDP solver was fixed to 9mins time-out to stop for generating the POMDP policy file with 2996 planes. The Fig 6.4 presents a example of one plane with corresponding action.

```
planes => [
  {
    action => 8,
    numEntries => 7,
    entries => [
      28, 8.58112,
      44, 9.58112,
      46, 8.58112,
      140, 8.63225,
      158, 8.63225,
      196, -3.65257,
      214, -3.65257
    ]
  },
```

Figure 6.4: One Plane Example of POMDP Policy File

The knowledge base which specifies the domain specific constraints can be encoded in any format of knowledge representation such as Ontology, frame, etc. Here, the basic knowledge representation format, rule, is utilized to implement the domain specific constraint knowledge based. In this rule based knowledge base, it encodes the netbook cannot be dual core and the laptop with dual core cannot be at the price of $600. For the belief state validation against knowledge based, the process is as following: at each time step, the belief states which have the possibility bigger than 0.05 are extracted. With the extracted the belief states, they will be validated against the rule based domain specific knowledge based to resolve the uncertainties existed in them. Only without any conflicts detected, the validation will be considered as successful, otherwise the belief states roll back process will be triggered to generate the hint and question to require the further information.

6.3 Experiments and Results

During a dialogue, the agent may take 13 different actions labeled as, for example, hello, askcorenum, and asktype, for conversation in natural language. For the purpose of testing, three different scenarios are used to examine the performance of the original and the modified new POMDP-based approaches in cases with or without conflicts in user's responses.

In the first scenario, the user provides all the needed information without any conflicts, and the requested product is available in stock. Shown in Fig. 6.5 is the sequence of dialogue, and listed in Table I. are belief states and their values. In the table, belief states with possibility value lower than 0.001 are not shown. In addition, an s in the last three characters of the belief state means that information for the corresponding factor is specified, and n means information is still missing. The results show that the original POMDP-based approach works in the same way as the proposed new approach. They both reach the goal state correctly after three rounds of conversation, which is the configuration at the highest probability 0.93 for a single-core netbook with a price tag of $600.

```
S: Hello
U: core1
S: asktype
U: typeN
S: askprice
U: price600
S: core1typeNprice600
```

Figure 6.5: Dialogue Process in Normal Scenario

Round #	Belief States	Value
0	core1typeLprice600-null-nnn	0.12
	core1typeLprice1200-null-nnn	0.12
	core1typeNprice600-null-nnn	0.12
	core1typeNprice1200-null-nnn	0.12
	core2typeLprice600-null-nnn	0.12
	core2typeLprice1200-null-nnn	0.12
	core2typeNprice600-null-nnn	0.12
	core2typeNprice1200-null-nnn	0.12
1	core1typeLprice600-core1-snn	0.24
	core1typeLprice1200-core1-snn	0.24
	core1typeNprice600-core1-snn	0.24
	core1typeNprice1200-core1-snn	0.24
2	core1typeNprice600-typeN-ssn	0.47
	core1typeNprice1200-typeN-ssn	0.47
3	core1typeNprice600-price600-sss	0.93
	core2typeNprice600-price600-sss	0.02
	core1typeLprice600-price600-sss	0.01
	core1typeNprice600-price600-nss	0.01
	core2typeNprice600-price600-nss	0.01
	core1typeNprice1200-price1200-sss	0.01

Table 6.1: Belief States in Normal Scenario for Both Approaches

```
S: Hello
U: core2
S: asktype
U: typeN
S: askprice
U: price600
S: core2typeNprice600
```

(a) Dialogue Process for POMDP-based Approach

```
S: Hello
U: core2
S: asktype
U: typeN
S: This product doesn't exist. typeN
   conflicts with core2. asktype
U: typeL
S: askprice
U: price600
S: This product doesn't exist. price600
   conflicts with typeL, core2. askprice
U: price1200
S: core2typeLprice1200
```

(b) Dialogue Process for the New Approach

Figure 6.6: Dialogue Processes for the 2nd and 3rd Scenarios

Shown in Fig. 6.6 and Table II-III are the results for the second and third scenarios. In the experiment, the second scenario relates to the case when the user asks for dual core for a netbook at the price of $600, which is an unreasonable configuration as netbooks come with single core only. The third scenario relates to the further dialogue after the system explains the conflict and the user asks for a laptop with dual core at $600. At this time, the system has to explain the conflict due to product unavailability and to guide the user reaching a feasible goal. The POMDP-based approach fails in both scenarios as it simply keeps updating the belief state without validation.

Round #	Belief States	Value
0	core1typeLprice600-null-nnn	0.12
	core1typeLprice1200-null-nnn	0.12
	core1typeNprice600-null-nnn	0.12
	core1typeNprice1200-null-nnn	0.12
	core2typeLprice600-null-nnn	0.12
	core2typeLprice1200-null-nnn	0.12
	core2typeNprice600-null-nnn	0.12
	core2typeNprice1200-null-nnn	0.12
1	core2typeLprice600-core2-snn	0.24
	core2typeLprice1200-core2-snn	0.24
	core2typeNprice600-core2-snn	0.24
	core2typeNprice1200-core2-snn	0.24
2	core2typeNprice600-typeN-ssn	0.47
	core2typeNprice1200-typeN-ssn	0.47
3	core2typeNprice600-price600-sss	0.93
	core1typeNprice600-price600-sss	0.02
	core2typeLprice600-price600-sss	0.01
	core1typeNprice600-price600-nss	0.01
	core2typeNprice600-price600-nss	0.01
	core2typeNprice1200-price1200-sss	0.01

Table 6.2: Belief States for the POMDP-based Approach

The new approach, on the other hand, is able to handle the uncertainties and reach to a

Round #	Belief States	Value
0	core1typeLprice600-null-nnn	0.12
	core1typeLprice1200-null-nnn	0.12
	core1typeNprice600-null-nnn	0.12
	core1typeNprice1200-null-nnn	0.12
	core2typeLprice600-null-nnn	0.12
	core2typeLprice1200-null-nnn	0.12
	core2typeNprice600-null-nnn	0.12
	core2typeNprice1200-null-nnn	0.12
1	core2typeLprice600-core2-snn	0.24
	core2typeLprice1200-core2-snn	0.24
	core2typeNprice600-core2-snn	0.24
	core2typeNprice1200-core2-snn	0.24
2	core2typeLprice600-core2-snn	0.24
	core2typeLprice1200-core2-snn	0.24
	core2typeNprice600-core2-snn	0.24
	core2typeNprice1200-core2-snn	0.24
3	core2typeLprice600-typeL-ssn	0.47
	core2typeLprice1200-typeL-ssn	0.47
4	core2typeLprice600-typeL-ssn	0.47
	core2typeLprice1200-typeL-ssn	0.47
5	core2typeLprice1200-price1200-sss	0.93
	core1typeLprice1200-price1200-sss	0.02
	core2typeNprice1200-price1200-sss	0.01
	core1typeLprice1200-price1200-nss	0.01
	core2typeNprice1200-price1200-nss	0.01

Table 6.3: Belief States for the Proposed Approach

goal that satisfies the need of the user. Though the new approach of dialogue management takes two more rounds of conversation than the POMDP-based approach in Fig. 6.6, the new approach takes into consideration the dependence of configuration factors and provides the user with useful guidance during the service. The original POMDP-based approach tries to reach a goal with only the information provided by the user, without considering uncertainties caused by user's lack of domain knowledge and by real-life constraints.

Chapter 7

Conclusion And Future Work

In this thesis, the main dialogue management approaches are reviewed under the flight ticket issuing agent. Also the history information space theory is discussed and a thorough analysis of the major approaches of dialogue management approaches with the theory of information space reveals reasons for their problems. With the analysis, the problem of the original POMDP based approach is identified. The Markovian over the belief state in the dialogue management process is problematic because it loses some significant information needed for the decision making. Therefore, the POMDP-based approach applied in the dialogue management cannot detect uncertainties in the belief state which are caused by the domain knowledge constraints. Based on the theory, a modified approach is proposed to enable POMDP-based dialogue management to handle uncertainties in belief state itself. Experimental results demonstrate significant improvement by the new approach towards accurate recognition of user's intention. The advantage is more obvious when it comes with the scenario that user has lack of knowledge and provides unreasonable information to the agent.

Since when the user is asking for a help, she or he is always lack of the particular domain knowledge, thus the proposed modified approach can be applied to the practical project to provide better services to the human user. For the future work, active investigation is under way to include the changing trend of belief state in the process of planning for the construction of a real truthful, relevant, clear, and informative dialogue system. Also, another important direction is that to investigate the more practical model to solve the POMDP based approach scale up problem. When the domain is complicated, the states space of POMDP specification file can be really huge and the POMDP solution is computation prohibitive. The current active researches have already put lots of efforts in this area to design more practical framework and POMDP solution algorithm to speed up the approximate solution finding process.

Bibliography

[1] M. Aretoulaki and B. Ludwig. Automatondescriptions and theorem-proving: a marriage made in heaven. In *Proceedings of IJCAI99 Workshop on Knowledge and Reasoning in Practical Dialogue Systems*, pages 9–16. IJCAI, 1999.

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

[3] D. Bohus and A.I. Rudnicky. RavenClaw: Dialog management using hierarchical task decomposition and an expectation agenda. In *Eighth European Conference on Speech Communication and Technology*. ISCA, 2003.

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

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

[6] Trung H. Bui, Dennis Hofs, and Boris van Schooten. Dialogue specification parser. `http://wwwhome.cs.utwente.nl/~hofs/pomdp/`, May 2010.

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

[8] R. Catizone, A. Setzer, and Y. Wilks. State of the Art in Dialogue Management. *In: Deliverable D5.1 of COMIC Project, www.hcrc.ed.ac.uk/comic/documents/*, 2002.

[9] P. Cohen. Dialogue modelling. *1996): Survey of the State of the Art in Human Language Technology. Oregon Graduate Institute of Technology, http://www. cse. ogi. edu/CSLU/HLTsurvey*, 1996.

[10] P. C Constantinides, S. Hansma, C. Tchou, and A. I Rudnicky. A schema based approach to dialog control. In *Fifth International Conference on Spoken Language Processing*, volume 2, pages 409–412. ICSLP, 1998.

[11] N. Dahlbäck, A. Flycht-Eriksson, A. Jönsson, and P. Qvarfordt. An architechture for multi-modal natural dialogue systems. In *ESCA Tutorial and Research Workshop (ETRW) on Interactive Dialogue in Multi-Modal Systems*, 1999.

[12] M. Denecke and A. Waibel. Dialogue strategies guiding users to their communicative goals. In *Proc. Eurospeech*, volume 96, 1997.

[13] N. Fraser. Assessment of interactive systems. *Handbook on Standards and Resources for Spoken Language Systems (D. Gibbon, R. Moore and R. Winski, eds.)*, pages 564–615, 1997.

[14] A.M. Garcia-Serrano, P. Martinez, and J.Z. Hernandez. Using AI techniques to support advanced interaction capabilities in a virtual assistant for e-commerce. *Expert Systems with applications*, 26(3):413–426, 2004.

[15] D. Goddeau, H. Meng, J. Polifroni, S. Seneff, and S. Busayapongchai. A form-based dialogue manager for spoken language applications. In *Fourth International Confer-*

ence on Spoken Language Processing, volume 2, pages 701–704. Philadelphia,PA, 1996.

[16] M.A. Goodrich and A.C. Schultz. Human-robot interaction: A survey. *Foundations and Trends in Human-Computer Interaction*, 1(3):203–275, 2007.

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

[18] S. Keizer. Reasoning under uncertainty in natural language dialogue using Bayesian networks. 2003.

[19] LB Larsen and A. Baekgaard. Rapid prototyping of a dialog system using a generic dialog development platform. In *Proc ICSLP International Conference on Spoken Language Processing, Yokohama*, pages 919–922, 1994.

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

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

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

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

[24] L. Rothkrantz, R. van Vark, A. Peters, and N. Andeweg. Dialogue control in the alparon system. In *Text, Speech and Dialogue*, pages 159–182. Springer, 2000.

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

[26] A.I. Rudnicky, E. Thayer, P. Constantinides, C. Tchou, R. Shern, K. Lenzo, W. Xu, and A. Oh. Creating natural dialogs in the Carnegie Mellon Communicator system. In *Sixth European Conference on Speech Communication and Technology*. ESCA, 1999.

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

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

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

[30] T. Smith. ZMDP software for POMDP and MDP planning. `http://www.cs.cmu.edu/~trey/zmdp/`, May 2010.

[31] T. Smith and R. Simmons. Heuristic search value iteration for POMDPs. In *Proceedings of the 20th conference on Uncertainty in artificial intelligence*, pages 520–527. AUAI Press, 2004.

[32] T. Smith and R. Simmons. Point-based POMDP algorithms: Improved analysis and implementation. In *Proc. Uncertainty in Artificial Intelligence*, 2005.

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

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

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

[36] W. Ward and B. Pellom. The cu communicator system. 1999.

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

[38] J.D. Williams, P. Poupart, and S. Young. Partially observable Markov decision processes with continuous observations for dialogue management. *Recent Trends in Discourse and Dialogue*, pages 191–217, 2008.

[39] J.D. Williams and S. Young. Scaling POMDPs for dialog management with composite summary point-based value iteration (CSPBVI). In *AAAI Workshop on Statistical and Empirical Approaches for Spoken Dialogue Systems*, pages 37–42, 2006.

[40] J.H. Wright, A.L. Gorin, and A. Abella. Spoken language understanding within dialogs using a graphical model of task structure. In *Fifth International Conference on Spoken Language Processing*, volume 5. ECSLP, 1998.

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

[42] W. Xu and A.I. Rudnicky. Task-based dialog management using an agenda. In *ANLP/NAACL 2000 Workshop on Conversational systems-Volume 3*, page 47. Association for Computational Linguistics, 2000.

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

[44] S. Young, M. Gasic, S. Keizer, F. Mairesse, J. Schatzmann, B. Thomson, and K. Yu. The hidden information state model: a practical framework for POMDP-based spoken dialogue management. *Computer Speech & Language*, 24(2):150–174, 2010.

[45] B. Zhang, Q. Cai, J. Mao, E. Chang, and B. Guo. Spoken dialogue management as planning and acting under uncertainty. In *Seventh European Conference on Speech Communication and Technology*, 2001.

Vita Auctoris

Libian Bian was born in Shanxi, China in 1991. She received his bachelor degree in software engineering at Beijing Institute of Technology in 2008. Currently, she is completing her master in computer science at the University of Windsor and expects to graduate in Fall 2010.