Flexible Spoken Dialogue System based on User Models and Dynamic Generation of VoiceXML Scripts

Kazunori Komatani Fumihiro Adachi Shinichi Ueno
Tatsuya Kawahara Hiroshi G. Okuno

Graduate School of Informatics
Kyoto University
Yoshida-Hommachi, Sakyo, Kyoto 606-8501, Japan
{komatani,adachi,ueno,kawahara,okuno}@kuis.kyoto-u.ac.jp

Abstract

We realize a telephone-based collaborative natural language dialogue system. Since natural language involves very various expressions, a large number of VoiceXML scripts need to be prepared to handle all possible input patterns. We realize flexible dialogue management for various user utterances by generating VoiceXML scripts dynamically. Moreover, we address appropriate user modeling in order to generate cooperative responses to each user. Specifically, we set up three dimensions of user models: *skill level* to the system, *knowledge level* on the target domain and the degree of *hastiness*. The models are automatically derived by decision tree learning using real dialogue data collected by the system. Experimental evaluation shows that the cooperative responses adapted to individual users serve as good guidance for novice users without increasing the dialogue duration for skilled users.

Keywords: spoken dialogue system, user model, VoiceXML, cooperative responses, dialogue strategy

1 Introduction

A Spoken dialogue system is one of the promising applications of the speech recognition and natural language understanding technologies. A typical task of spoken dialogue systems is database retrieval. Some IVR (interactive voice response) systems using the speech recognition technology are being put into practical use as its simplest form. According to the spread of cellular phones, spoken dialogue systems via telephone enable us to obtain information from various places without any other special apparatuses.

In order to realize user-friendly interaction, spoken dialogue systems should be able to (1) accept various user utterances to enable mixed-initiative dialogue and (2) generate cooperative responses. Currently, a lot of IVR systems via telephone operate by using VoiceXML, which is a script language to prescribe procedures of spoken dialogues. However, only the next behaviors corresponding to every input are prescribed in the VoiceXML scripts, so the dialogue procedure is basically designed as system-initiated one, in which the system asked required items one by one. In order to realize mixed-initiative dialogue, the system should be able to accept various user-initiated utterances. By allowing to accept various user utterances, the combination of words included in the utterances accordingly gets enormous, and then it is practically impossible to prepare VoiceXML scripts that correspond to the enormous combinations of input words in advance. It is also difficult to generate cooperative responses adaptively in the above framework.

We propose a framework to generate VoiceXML scripts dynamically in order to realize the mixed-initiative dialogue, in which the system is needed to accept various user utterances. This framework realizes flexible dialogue management without requiring

preparation for a large number of VoiceXML scripts in advance. Furthermore, it enables various behaviors adaptive to the dialogue situations such as obtained query results.

Another problem to realize user-friendly interaction is how to generate cooperative responses. When we consider the responses generated from the system side, the dialogue strategies, which determine when to make guidance and what the system should tell to the user, are the essential factors in spoken dialogue systems. There are many studies in respect of the dialogue strategy such as confirmation management using confidence measures of speech recognition results (Komatani and Kawahara, 2000; Hazen et al., 2000), dynamic change of dialogue initiative (Litman and Pan, 2000; Chu-Carroll, 2000; Lamel et al., 1999), and addition of cooperative contents to system responses (Sadek, 1999). Nevertheless, whether a particular response is cooperative or not depends on individual user's characteristic.

In order to adapt the system's behavior to individual users, it is necessary to model the user's patterns (Kass and Finin, 1988). Most of conventional studies on user models have focused on the knowledge of users. Others tried to infer and utilize user's goals to generate responses adapted to the user (van Beek, 1987; Paris, 1988). Elzer et al. (2000) proposed a method to generate adaptive suggestions according to users' preferences. However, these studies depend on knowledge of the target domain greatly, and therefore the user models need to be deliberated manually to be applied to new domains. Moreover, they assumed that the input is text only, which does not contain errors.

We propose more comprehensive user models to generate user-adapted responses in spoken dialogue systems taking account of information specific to spoken dialogue. Spoken utterances include various information such as the interval between the utterances, the presence of barge-in and so on, which can be utilized to judge the user's character. These features also possess generality in spoken dialogue systems because they are not dependent on domain-specific knowledge. As user models in spoken dialogue systems, Eckert et al. (1997) defined stereotypes of users such as patient, submissive and experienced, in order to evaluate spoken dialogue systems by simulation. We introduce user models not for defining users' behaviors beforehand, but for detecting users' patterns in real-time interaction.

We define three dimensions in the user models: '*skill level* to the system', '*knowledge level* on the target domain' and 'degree of *hastiness*'. The user models are trained by decision tree learning algorithm, using real data collected from the Kyoto city bus information system. Then, we implement the user models on the system and evaluate them using data collected with 20 novice users.

2 Flexible Spoken Dialogue System based on Dynamic Generation of VoiceXML Scripts

VoiceXML[1] is a script language to prescribe procedures in spoken dialogues mainly on telephone, and is becoming to a standard language of IVR systems. The VoiceXML scripts consist of three parts: (1) specifications of system's prompts, (2) specifications of grammars to accept a user's utterance, and (3) description of the next behaviors.

However, most of existing services using the VoiceXML imposes rigid interaction, in which user utterances are restricted by system-initiated prompts and a user is accordingly allowed to specify only requested items one by one. It is more user-friendly that users can freely convey their requests by natural language expressions.

We present a framework to realize flexible interaction by generating VoiceXML scripts dynamically (Pargellis et al., 1999; Nyberg et al., 2002). The framework enables users to express their requests by natural language even in VoiceXML-based systems. Furthermore, cooperative responses in the Kyoto city bus information system that has been developed in our laboratory are also presented in this section.

2.1 Dynamic Generation of VoiceXML Scripts

In VoiceXML scripts, acceptable keywords and corresponding next states must be explicitly specified. However, since there exists enormous combinations of keywords in natural language expressions, it is practically impossible to describe all VoiceXML scripts that correspond to the combinations. Then, we introduce the framework in which VoiceXML

[1] VoiceXML Forum. `http://www.voicexml.org/`

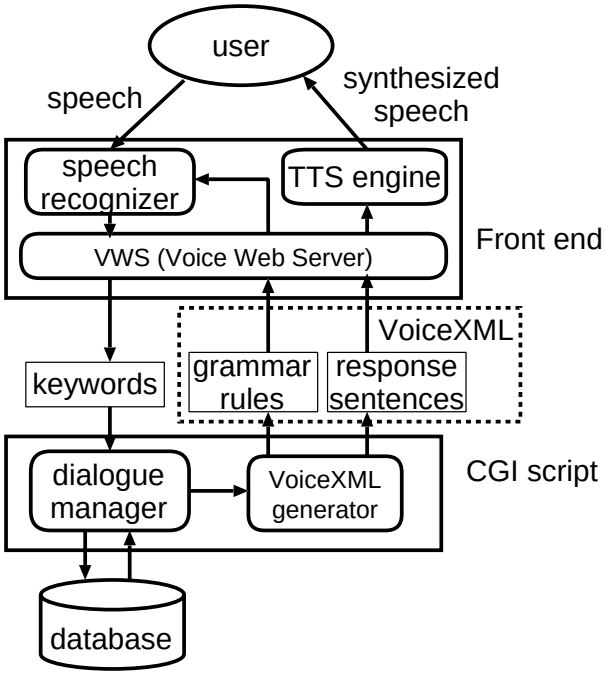

Figure 1: Overview of spoken dialogue system based on dynamic generation of VoiceXML scripts

scripts are generated dynamically to enable the system to accept natural language expressions.

Figure 1 shows the overview of the framework. The front end that operates based on VoiceXML scripts is separated from the dialogue management portion, which accepts speech recognition results and generates corresponding VoiceXML scripts. The user utterance is recognized based on a grammar rule specified in VoiceXML scripts, and keywords extracted from a speech recognition result are passed to a CGI script. The CGI script retrieves corresponding information from the database on Web, and generates a VoiceXML script for the succeeding interaction. If sufficient information is not obtained from a user utterance, a script that prompts to fill remaining contents is generated, and if the user utterance contains ambiguity, a script that makes a disambiguating question is generated.

Consequently, the generation of VoiceXML scripts enables to accept natural language expressions without preparing a large number of the scripts corresponding to various inputs beforehand. The framework also enables to generate cooperative responses adapted to the situation such as retrieval results without spoiling portability.

Sys: Please tell me your current bus stop, your destination or the specific bus route.
User: Shijo-Kawaramachi.
Sys: Do you take a bus from Shijo-Kawaramachi?
User: Yes.
Sys: Where will you get off the bus?
User: Arashiyama.
Sys: Do you go from Shijo-Kawaramachi to Arashiyama?
User: Yes.
Sys: Bus number 11 bound for Arashiyama has departed Sanjo-Keihanmae, two bus stops away.

Figure 2: Example dialogue of the bus system

2.2 Kyoto City Bus Information System

We have developed the Kyoto city bus information system, which locates the bus a user wants to take, and tell him/her how long it will take before arriving. The system can be accessed via telephone including cellular phones[2]. From any places, users can easily get the information on the bus that changes every minute. Users are requested to input the bus stop to get on, the destination, or the bus route number by speech, and consequently get the bus information. There are 652 bus stops and 85 bus routes in Kyoto city. The bus stops can also be specified by the name of famous places or public facilities nearby. Figure 2 shows a simple example of the dialogue.

2.3 Generation of Cooperative Responses in Kyoto City Bus Information System

In spoken dialogue systems, it is important to generate cooperative responses by guessing the user's intention. The cooperative responses consist of adding information more than the user has requested, telling the reason when there is no query result, presenting the alternatives, and so on (Sadek, 1999). In order to generate cooperative responses, it is necessary to change responses according to the situations. Since we separate language-understanding modules from individual VoiceXML scripts that are generated dynamically, our system can generate appropriate responses cooperatively according to query results. The examples of the cooperative responses implemented in our Kyoto city bus information system are listed below.

[2] +81-75-326-3116

(1) Suggestive answers
The information related to a query result is offered when the query result is negative.
(Example)

User: From Hyakumanben to Kyoto Station by bus number 17.
Sys: Bus number 17 does not come for a while, *but bus number 206 has departed Takano, four bus stops away*.

(2) Corrective answers
The reason is presented when there is no query result or when a user utterance contains an error.
(Example)

User: From Kyoto Station to Arashiyama.
Sys: *There is no bus that directly goes from Kyoto station to Arashiyama*.

(3) Intensional answers
If several query results are obtained, generating intensional answer is more cooperative than listing all the results.
(Example)

User: From Shijo-Kawaramachi to Shijo-Karasuma.
Sys: *All buses* to Shijo-Karasuma do not come for a while.

(4) Completion answers
It is cooperative to generate responses including information more than the user's request. However, whether the information should be presented or not differs in individual users.
(Example)

User: From Kyoto station by bus number 5.
Sys: Bus number 5 will depart at 17:34 *from platform 'A1'*.
(There are a lot of platforms in Kyoto station.)

3 Response Generation Using User Models

We propose comprehensive user models to generate user-adaptive responses taking account of information specific to spoken dialogue. Whether a particular response is regarded as cooperative depends on individual user's characteristics. So, we address appropriate user modeling in order to generate cooperative responses to the users.

3.1 Classification of User Models

We define three dimensions as user models listed below.

- Skill level to the system
- Knowledge level on the target domain
- Degree of hastiness

Skill Level to the System

Since spoken dialogue systems are not widespread yet, there arises a difference in the skill level of users in operating the systems. It is desirable that the system changes its behavior including response generation and initiative management in accordance with the skill level of the user. In conventional systems, a system-initiated guidance has been invoked on the spur of the moment either when the user says nothing or when speech recognition is not successful. In our framework, we address a radical solution for the unskilled users by modeling the skill level as the user's property before such a problem arises.

Knowledge Level on the Target Domain

There also exists a difference in the knowledge level on the target domain among users. Thus, it is necessary for the system to change information to present to users. For example, it is not cooperative to tell too detailed information to strangers. On the other hand, for inhabitants, it is useful to omit too obvious information and to output more detailed information. Therefore, we introduce a dimension that represents the knowledge level on the target domain.

Degree of Hastiness

In speech communications, it is more important to present information promptly and concisely compared with the other communication modes such as browsing. Especially in the bus system, the conciseness is preferred because the bus information is urgent to most users. Therefore, we also take account of degree of hastiness of the user, and accordingly change the system's responses.

3.2 Response Generation Strategy Using User Models

Next, we describe the response generation strategies adapted to individual users based on the proposed

user models: *skill level*, *knowledge level* and *hastiness*. Basic design of dialogue management is based on mixed-initiative dialogue, in which the system makes follow-up questions and guidance if necessary while allowing a user to utter freely. It is investigated to add various contents to the system responses as cooperative responses in (Sadek, 1999). Such additional information is usually cooperative, but some people may feel such a response redundant.

Thus, we introduce the user models and control the generation of additional information. By introducing the proposed user models, the system changes generated responses by the following two aspects: dialogue procedure and contents of responses.

Dialogue Procedure

The dialogue procedure is changed based on the *skill level* and the *hastiness*. If a user is identified as having the high *skill level*, the dialogue management is carried out in a user-initiated manner; namely, the system generates only open-ended prompts. On the other hand, when user's *skill level* is detected as low, the system takes an initiative and prompts necessary items in order.

When the degree of *hastiness* is low, the system makes confirmation on the input contents. Conversely, when the *hastiness* is detected as high, such a confirmation procedure is omitted; namely, the system immediately makes a query and outputs the result without making such a confirmation.

Contents of Responses

Information that should be included in the system response can be classified into the following two items.

1. Dialogue management information
2. Domain-specific information

The dialogue management information specifies how to carry out the dialogue including the instruction on user's expression for yes/no questions like "Please reply with either yes or no." and the explanation about the following dialogue procedure like "Now I will ask in order." This dialogue management information is determined by the user's *skill level* to the system, and is added to system responses when the *skill level* is considered as low.

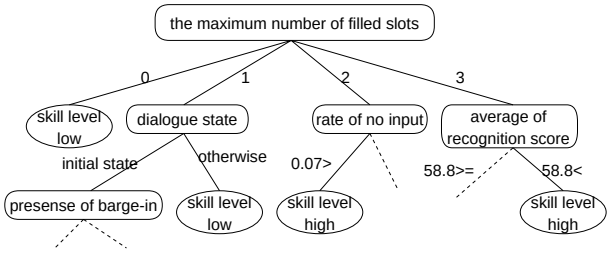

Figure 3: Decision tree for the *skill level*

The domain-specific information is generated according to the user's *knowledge level* on the target domain. Namely, for users unacquainted with the local information, the system adds the explanation about the nearest bus stop, and omits complicated contents such as a proposal of another route.

The contents described above are also controlled by the *hastiness*. For users who are not in hurry, the system generates the additional contents that correspond to their *skill level* and *knowledge level* as cooperative responses. On the other hand, for hasty users, the contents are omitted to prevent the dialogue from being redundant.

3.3 Classification of User based on Decision Tree

In order to implement the proposed user models as classifiers, we adopt a decision tree. It is constructed by decision tree learning algorithm C5.0 (Quinlan, 1993) with data collected by our dialogue system. Figure 3 shows an example of the derived decision tree for the *skill level*.

We use the features listed in Figure 4. They include not only semantic information contained in the utterances but also information specific to spoken dialogue systems such as the silence duration prior to the utterance, the presence of barge-in and so on. Except for the last category of Figure 4 including "attribute of specified bus stops", most of the features are domain-independent.

The classification of each dimension is done for every user utterance except for *knowledge level*. The model of a user can change during a dialogue. Features extracted from utterances are accumulated as history information during the session.

- features obtained from a single utterance
  - dialogue state (defined by already filled slots)
  - presence of barge-in
  - lapsed time of the current utterance
  - recognition result (something recognized / uncertain / no input)
  - score of speech recognizer
  - the number of filled slots by the current utterance
- features obtained from the session
  - the number of utterances
  - dialogue state of the previous utterance
  - lapsed time from the beginning of the session
  - the number of the repetition of a same question
  - the average number of the repetition of a same question
  - ratio of the time of user utterances in whole elapsed time
  - ratio of the occurrence of barge-in out of the whole number of utterances
  - recognition result of the previous utterance
  - ratio of something recognized
  - ratio of getting uncertain results
  - ratio of no input
  - the number of barge-in
  - the number of something recognized
  - the number of getting uncertain results
  - the number of no input
  - average of recognition scores
  - the maximum number of slots filled by a single utterance
- features obtained from the session (used only in classification of *knowledge level*)
  - attribute of specified bus stops
  - the way of specifying bus stops (whether a bus stop is specified by its correct name or not)
  - ratio of each attribute of specified bus stops
  - ratio of each way of specifying bus stops

Figure 4: Features to classify the user models

case: [skill level: low][knowledge level: low] [hastiness: low]

Sys: Please tell me your current bus stop, your destination or the specific bus route number.
User: Maruyama Park.
Sys: Let me confirm one by one.
(← The explanation is added by [skill level: low].)
Will you get on at Maruyama Park?
(← The content is confirmed by [hastiness: low].)
Please reply with yes or no.
(← The instruction is added by [skill level: low].)
User: Yes.
Sys: Where will you get off the bus? Please specify by a bus stop name or a well-known landmark.
(← The explanation is added by [skill level: low].)
User: Tofukuji temple.
Sys: Will you get off the bus at Tofukuji temple?
Please reply with yes or no.
User: Yes.
Sys: The following buses are located.
(← The explanation is added by [skill level: low].)
The nearest bus stop to Maruyama Park is Gion.
(← The domain information is added by [knowledge level: low].)
Bus number 202 bound for Nishioji-Kujo has departed Higashiyama-Sanjo, which is two stops away.
...

Figure 5: An example dialogue with the proposed user models

	low	indeterminable	high	total
skill level	743	253	496	1492
knowledge level	275	808	409	1492
hastiness	421	932	139	1492

Table 1: Number of manually annotated labels for decision tree learning

Figure 5 shows an example of the system behavior with the proposed user models. The *skill level* is classified as being low by the decision tree, because the first user's utterance includes only one content word. Then, dialogue procedure is changed to the system-initiated one. Similarly, the *hastiness* is classified as being low by the decision tree, and the system includes the explanation on the dialogue procedure and the instruction on the expression in the responses. They are omitted if the *hastiness* is identified as high.

3.4 Decision Tree Learning for User Models

We train and evaluate the decision tree for the user models using dialogue data collected by our system. The data was collected from December 10th 2001 to May 10th 2002. The number of the sessions (telephone calls) is 215, and the total number of utterances included in the sessions is 1492. We annotated the subjective labels of the user models by hand. The annotator judges the user models for every utterance based on the recorded speech data and logs. The labels were given to the three dimensions described in section 3.1 among 'high', 'indeterminable' or 'low'. It is possible that the annotated model of a user changes during a dialogue, especially from 'indeterminable' to 'low' or 'high'. The number of the labeled utterances is shown in Table 1.

condition	#1	#2	#3
skill level	80.8%	75.3%	85.6%
knowledge level	73.9%	63.7%	78.2%
hastiness	74.9%	73.7%	78.6%

Table 2: Classification accuracy of the proposed user models

Using the labeled data, we trained the decision tree and evaluated the classification accuracy of the proposed user models. All the experiments were carried out by the method of 10-fold cross validation. The process, in which one tenth of all data is used as the test data, and the remainder is used as the training data, is repeated ten times, and the average of the classification accuracy is computed. The result is shown in Table 2. The conditions #1, #2 and #3 in Table 2 are described as follows.

#1: The 10-fold cross validation is carried out per utterance.

#2: The 10-fold cross validation is carried out per session (call).

#3: We calculate the accuracy under more realistic condition. The accuracy is calculated not in three classes (high / indeterminable / low) but in two classes that actually affect the dialogue strategies. For example, the accuracy for the *skill level* is calculated for the two classes: low and the others. As to the classification of *knowledge level*, the accuracy is calculated for dialogue sessions, because the features such as the attribute of a specified bus stop are not obtained in every utterance. Moreover, in order to smooth unbalanced distribution of the training data, a cost corresponding to the reciprocal ratio of the number of samples in each class is introduced. By the cost, the chance rate of two classes becomes 50%.

The difference between condition #1 and #2 is that the training was carried out in a speaker-closed or speaker-open manner. The former shows better performance.

The result in condition #3 shows useful accuracy in the *skill level*. The following features play important part in the decision tree for the *skill level*: the number of filled slots by the current utterance, presence of barge-in and ratio of no input. For the *knowledge level*, recognition result (something recognized / uncertain / no input), ratio of no input and the way to specify bus stops (whether a bus stop is specified by its exact name or not) are effective. The *hastiness* is classified mainly by the three features: presence of barge-in, ratio of no input and lapsed time of the current utterance.

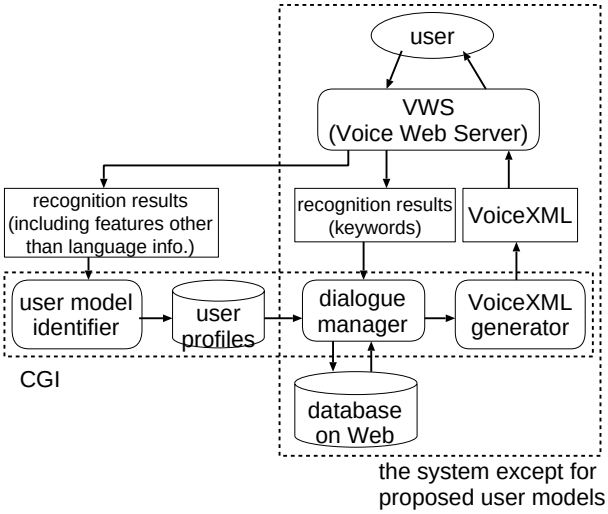

Figure 6: Overview of the Kyoto city bus information system with user models

3.5 System Overview

Figure 6 shows an overview of the Kyoto city bus information system with the user models. The system operates by generating VoiceXML scripts dynamically as described in section 2.1. The real-time bus information database is provided on the Web, which can be accessed via Internet. Then, we explain the modules in the following.

VWS (Voice Web Server)

The Voice Web Server drives the speech recognition engine and the TTS (Text-To-Speech) module accordingly to the specifications by the generated VoiceXML script.

Speech Recognizer

The speech recognizer decodes user utterances based on specified grammar rules and vocabulary, which are defined by VoiceXML at each dialogue state.

Dialogue Manager

The dialogue manager generates response sentences based on recognition results (bus stop names or a route number) received from the VWS. If sufficient information to locate a bus is obtained, it retrieves the corresponding bus information on the Web.

VoiceXML Generator

This module dynamically generates VoiceXML scripts that contain response sentences and specifications of speech recognition grammars, which are given by the dialogue manager.

User model identifier

This module classifies user's characters based on the user models using features specific to spoken dialogue as well as semantic attributes. The obtained user profiles are sent to the dialogue manager, and are utilized in the dialogue management and response generation.

4 Experimental Evaluation of the System with User Models

We evaluated the system with the proposed user models using 20 novice subjects who had not used the system. The experiment was performed in the laboratory under adequate control. For the speech input, the headset microphone was used.

4.1 Experiment Procedure

First, we explained the outline of the system to subjects and gave a document in which experiment conditions and scenarios were described. We prepared two sets of eight scenarios. Subjects were requested to acquire the bus information according to the scenarios using the system with/without the user models. In the scenarios, neither the concrete names of bus stops nor the bus number were given. For example, one of the scenarios was as follows: "You are in Kyoto for sightseeing. After visiting the Ginkakuji temple, you go to Maruyama Park. Supposing such a situation, please get information on the bus." We also set the constraint in order to vary the subjects' hastiness such as "Please hurry as much as possible in order to save the charge of your cellular phone."

The subjects were also told to look over questionnaire items before the experiment, and filled in them after using each system. This aims to reduce the subject's cognitive load and possible confusion due to switching the systems (Over, 1999). The questionnaire consisted of eight items, for example, "When the dialogue did not go well, did the system guide intelligibly?" We set seven steps for evaluation about each item, and the subject selected one of them.

Furthermore, subjects were asked to write down the obtained information: the name of the bus stop to get on, the bus number and how much time it takes before the bus arrives. With this procedure, we planned to make the experiment condition close to the realistic one.

The subjects were divided into two groups; a half (group 1) used the system in the order of "with user models → without user models", the other half (group 2) used in the reverse order.

The dialogue management in the system without user models is also based on the mixed-initiative dialogue. The system generates follow-up questions and guidance if necessary, but behaves in a fixed manner. Namely, additional cooperative contents corresponding to *skill level* described in section 3.2 are not generated and the dialogue procedure is changed only after recognition errors occur. The system without user models behaves equivalently to the initial state of the user models: the *hastiness* is low, the *knowledge level* is low and the *skill level* is high.

		duration (sec.)	# turn
group 1	with UM	51.9	4.03
(with UM → w/o UM)	w/o UM	47.1	4.18
group 2	w/o UM	85.4	8.23
(w/o UM → with UM)	with UM	46.7	4.08

UM: User Model

Table 3: Duration and the number of turns in dialogue

4.2 Results

All of the subjects successfully completed the given task, although they had been allowed to give up if the system did not work well. Namely, the task success rate is 100%.

Average dialogue duration and the number of turns in respective cases are shown in Table 3. Though the users had not experienced the system at

group 1	with UM	0.72
(with UM → w/o UM)	w/o UM	0.70
group 2	w/o UM	0.41
(w/o UM → with UM)	with UM	0.63

Table 4: Ratio of utterances for which the skill level was judged as high

all, they got accustomed to the system very rapidly. Therefore, as shown in Table 3, the duration and the number of turns were decreased obviously in the latter half of the experiment in both groups. However, in the initial half of the experiment, the group 1 completed with significantly shorter dialogue than group 2. This means that the incorporation of the user models is effective for novice users. Table 4 shows a ratio of utterances for which the skill level was identified as high. The ratio is calculated by dividing the number of utterances that were judged as high skill level by the number of all utterances. The ratio is much larger for group 1 who initially used the system with user models. This fact means that the novice users got accustomed to the system more rapidly with the user models, because they were instructed on the usage by cooperative responses generated when the *skill level* is low. The results demonstrate that cooperative responses generated according to the proposed user models can serve as good guidance for novice users.

In the latter half of the experiment, the dialogue duration and the number of turns were almost same between the two groups. This result shows that the proposed models prevent the dialogue from becoming redundant for skilled users, although generating cooperative responses for all users made the dialogue verbose in general. It suggests that the proposed user models appropriately control the generation of cooperative responses by detecting characters of individual users.

5 Conclusions

We have presented a framework to realize flexible interaction by dynamically generating VoiceXML scripts. This framework realizes mixed-initiative dialogues and the generation of cooperative responses in VoiceXML-based systems.

We have also proposed and evaluated user models for generating cooperative responses adaptively to individual users. The proposed user models consist of the three dimensions: *skill level* to the system, *knowledge level* on the target domain and the degree of *hastiness*. The user models are identified by decision tree using features specific to spoken dialogue systems as well as semantic attributes. They are automatically derived by decision tree learning, and all features used for *skill level* and *hastiness* are independent of domain-specific knowledge. So, it is expected that the derived user models can be generally used in other domains.

The experimental evaluation with 20 novice users shows that the skill level of novice users was improved more rapidly by incorporating user models, and accordingly the dialogue duration becomes shorter more immediately. The result is achieved by the generated cooperative responses based on the proposed user models. The proposed user models also suppress the redundancy by changing the dialogue procedure and selecting contents of responses.

Thus, the framework generating VoiceXML scripts dynamically and the proposed user models realize a user-adaptive dialogue strategies, in which the generated cooperative responses serve as good guidance for novice users without increasing the dialogue duration for skilled users.

References

Jennifer Chu-Carroll. 2000. MIMIC: An adaptive mixed initiative spoken dialogue system for information queries. In *Proc. of the 6th Conf. on applied Natural Language Processing*, pages 97–104.

Wieland Eckert, Esther Levin, and Roberto Pieraccini. 1997. User modeling for spoken dialogue system evaluation. In *Proc. IEEE Workshop on Automatic Speech Recognition and Understanding*, pages 80–87.

Stephanie Elzer, Jennifer Chu-Carroll, and Sandra Carberry. 2000. Recognizing and utilizing user preferences in collaborative consultation dialogues. In *Proc. of the 4th Int'l Conf. on User Modeling*, pages 19–24.

Timothy J. Hazen, Theresa Burianek, Joseph Polifroni, and Stephanie Seneff. 2000. Integrating recognition confidence scoring with language understanding and dialogue modeling. In *Proc. Int'l Conf. Spoken Language Processing (ICSLP)*.

Robert Kass and Tim Finin. 1988. Modeling the user in natural language systems. *Computational Linguistics*, 14(3):5–22.

Kazunori Komatani and Tatsuya Kawahara. 2000. Flexible mixed-initiative dialogue management using concept-level confidence measures of speech recognizer output. In *Proc. Int'l Conf. Computational Linguistics (COLING)*, pages 467–473.

Lori Lamel, Sophie Rosset, Jean-Luc Gauvain, and Samir Bennacef. 1999. The LIMSI ARISE system for train travel information. In *IEEE Int'l Conf. Acoust., Speech & Signal Processing (ICASSP)*.

Diane J. Litman and Shimei Pan. 2000. Predicting and adapting to poor speech recognition in a spoken dialogue system. In *Proc. of the 17th National Conference on Artificial Intelligence (AAAI2000)*.

Eric Nyberg, Teruko Mitamura, Paul Placeway, Michael Duggan, and Nobuo Hataoka. 2002. Dialogxml: Extending voicexml for dynamic dialog management. In *Proc. of Human Language Technology 2002 (HLT2002)*, pages 286–291.

Paul Over. 1999. Trec-7 interactive track report. In *Proc. of the 7th Text REtrieval Conference (TREC7)*.

Andrew Pargellis, Jeff Kuo, and Chin-Hui Lee. 1999. Automatic dialogue generator creates user defined applications. In *Proc. European Conf. Speech Commun. & Tech. (EUROSPEECH)*.

Cecile L. Paris. 1988. Tailoring object descriptions to a user's level of expertise. *Computational Linguistics*, 14(3):64–78.

J. Ross Quinlan. 1993. *C4.5: Programs for Machine Learning.* Morgan Kaufmann, San Mateo, CA. http://www.rulequest.com/see5-info.html.

David Sadek. 1999. Design considerations on dialogue systems: From theory to technology -the case of artimis-. In *Proc. ESCA workshop on Interactive Dialogue in Multi-Modal Systems*.

Peter van Beek. 1987. A model for generating better explanations. In *Proc. of the 25th Annual Meeting of the Association for Computational Linguistics (ACL-87)*, pages 215–220.