DYNAMIC AND GOAL-ORIENTED INTERACTION
FOR MULTI-MODAL SERVICE AGENTS

Tommy SHEU
Integration Technology Laboratory,
Institute for Information Industry, Taipei
tommy@itl.iii.org.tw

Bor-Shen LIN
Integration Technology Laboratory,
Institute for Information Industry, Taipei
bslin@itl.iii.org.tw

ABSTRACT
Form-based dialogue modeling schemes such as VoiceXML
specification have been widely used in designs of dialogue
management. However, for such schemes the dialogue goals have
to be rigidly modeled as form-filling problems, while the data
presentation needs to be statically defined in the forms
beforehand. This not only limits the application scope of
dialogue systems to form-filling tasks, but also constrains the
versatility and variety for human computer interactions. In this
paper, a dialogue management approach providing dynamic and
goal-oriented interaction is proposed. This approach uses the
event hierarchy to represent the problem-solving procedures, as
in conventional plan-based schemes. However, instead of using
the logic programming languages, it constructs the logical
relationships among dialogue goals, conditions and objects with
XML-tree structure, while using ECMA-script to perform
procedural computation, which make this scheme superior in
capabilities of object representation and procedural computation.
The next action obtained from inferring on the XML-tree is
mode-independent, which can be used to generate the
presentation dynamically with multiple modes, including speech,
text, GUI or expression of talking head. This proposed scheme
has been applied to a multi-modal dialogue agent for personal
information services providing goal-oriented interaction.

1. INTRODUCTION
Recently, Microsoft and several speech companies proposed a
new specification named SALT for multi-modal dialogue
systems [1]. This specification was not defined from the scratch
but as an extension of the widely used HTML standard. Based on
this specification, a system can accept the input speech of a user,
while the interpretative results can change the form values of
web documents. This is really helpful for quickly deploying
dialogue systems just like VoiceXML specification [2]. However,
with the implicit assumptions of form-based dialogue modeling,
such specifications are not flexible enough for supporting the
scenarios that cannot be easily formulated as form-filling
problems. Also, due to strict bind to static HTML form, it is not
easy for such specification to support dynamic and goal-oriented
interaction with humans. Thus, in this paper, a dialogue
management approach extended from previous research [3] is
proposed and described below.

In conventional plan-based dialogue modeling schemes, event
hierarchy [3-6] is used to describe the logic relationships among
dialogue goals, conditions and objects, and usually constructed
with logic programming languages. However, the logic
conditions are often quite dynamic and sometimes need to be
determined by procedural computation that cannot be well
handled by logic programming languages. Also, the logic
programming languages are weaker in the capability of object
representation than procedural languages, which is very critical
for handling the dialogue state. To solve this problem, a dialogue
management approach extended from a plan-based modeling
scheme in earlier research [3] is proposed. This approach
remains the capability of handling event hierarchy (as in
symbolic AI languages) by constructing the logical relationships
among dialogue goals, conditions and objects with/or XML-
tree structure, while enhancing the capabilities of procedural
computation and object representation through ECMA-script
language. The fundamental difference between this approach and
form-based modeling schemes (such as SALT or VXML) is that
the dialogue control for this approach is based on ‘object’ and
‘logic’ instead of ‘form’ and ‘field’, and thus can be generalized
and applied to non-form-filling tasks more easily. This approach
has been applied to a multi-modal dialogue agent for personal
information services.

The rest of this paper is organized as follows. The system
architecture for general multi-modal human computer interaction
and a simplified diagram are depicted in Section 2. An
application example for PIM is used for illustration in Section 3.
Software technologies for integration are briefly described in
Section 4. Finally some conclusions are made in Section 5.

2. SYSTEM ARCHITECTURE

2.1 System architecture for multi-modal HCI
The service agent consists of three layers: device interface layer,
message interpretation layer and interaction control layer. Figure
1 shows the system architecture for general multi-modal HCI.
- Device interface layer
Multi-modal inputs include acoustic interface, keyboard, GUI interface, and camera. Multi-modal outputs include graph/form, text, talking head and speech. User input could be natural speech, finger touch or text, and is converted to universal semantic representation. System output includes voice, graph, table and text, which are presented dynamically according to current state.

- Message interpretation layer
Message interpretation layer includes speech recognition, natural language processing, message interpreter, message dispatcher, language generation and speech synthesis.

- Interaction control layer
Interaction control layer includes dialogue manager and information server. Information servers provide different services to dialogue manager. They may be personal information management, lab resource or traffic status. The information source comes from database, file system or Internet resource. Either SQL-based or content-based IR method could be applied to retrieve structural or non-structural information.

2.2 Simplified diagram for multi-modal HCI

For simplicity, the system architecture depicted in Figure 1 can be divided into four stages: Perception, Update, Inference and Presentation, as shown in Figure 2. Perception includes multi-modal input of device interface layer and speech recognition, natural language processing, message interpreter of message interpretation layer. Dialogue manager of interaction control layer is divided into Update and Inference. The domain-independent part of dialogue manager can load and interpret domain-dependent XML documents in which the event hierarchy of dialogue goals is defined. Presentation includes multi-media output of device interface layer and message dispatcher, language generation, speech synthesis of message interpretation layer.

Figure 1. System architecture for multi-modal HCI.

Figure 2. Simplified diagram for multi-modal HCI.

- Perception
Perception is gotten from multi-modal inputs such as natural speech, finger touch, or text input, and converted into universal semantic representation. For example, if a user says help, types help text or click help button, the system will receive the same semantic slot representing that the user needs help information. Any input event will trigger the cycle of ‘Perception→Update→Inference→Presentation’.

- Update
There are three states for each semantic slot: unknown, known and verified [3]. Slots are set, verified, changed or cleared according to the perception result. Different perceptions are processed using different strategies to update dialogue status. For natural speech, keywords are recognized, user's intention is understood and needed to be further confirmed because of possible recognition error. For finger touch, selection from the user is clear and verified directly. For text input, whether the confirmation is needed or not just depends.

- Inference
Dialogue management constructs the logical relationships among objects, conditions and goals with XML-tree structure describing the event hierarchy, while using ECMA-script to perform procedural computation, which make this scheme superior in capabilities of object description and procedural computation. For multi-topic control, topic is resolved, cross-topic data is inherited and dialogue history is recorded for topic switching [3]. Inference is proceeded from the top of and/or XML dialogue control tree. An OR-type tree means that anyone is enough for the achievement of this sub-goal. An AND-type tree means that everyone must be finished to fulfill this sub-goal. The traversing path is depth-first for each tree, and continued till current tree is done. Focus is found from the tree when the expression of a certain node is false, and the corresponding action is performed. An example further illustrating the XML-tree in details will be given in Section 3.
- Presentation
Presentation consists of response strategy and response generation. Querying one of unknown slots and confirming all known/unverified slots is adopted in the response strategy. Response generation includes text, graph, table and voice of querying and confirming. Presentation is shown independently of inference and dynamically according to current state. A system converses with a user in appropriate reaction, and only needed information will be presented on the screen. This avoids too many irrelevant items occurring at the same time, making no focus of choices to the user.

3. AN APPLICATION EXAMPLE

With the integration of ASR, TTS, DM, NLP, face detection and talking head, dynamic and goal-oriented interaction for multimodal service agents are achieved as below.

3.1 PIM query’s scenario

(A user walks to the PIM service agent system.)
System: What may I help you?
User: I want to get my calendar.
System: What date do you want to look up?
User: This afternoon.
System: What time do you want to look up this afternoon?
(The user selects 4 o’clock button in the form)
System: There are two appointments at 4 o’clock this afternoon. The first one is ‘advanced research technology project’…
User: Thanks.
System: You are welcome. What else may I help you?
User: Is there any email about ‘advanced research technology project’?
System: Yes, there is one relevant email found from Bor-Shen…
User: Tell me his email address.
System: Bor-Shen’s email address is bslin@itl.iii.org.tw. Anything else?
User: No, thank you.

3.2 Goal-Oriented Inference

An XML dialogue control tree consisting of event hierarchy as described in Section 2.2 is applied to inferring user’s intention. Figure 3 shows the OR/AND tree structure of PIM query. OR-type tree of ‘goal’ contains five AND-type sub-trees (calendar, contact, email, todo, note), and each sub-tree has several cond-type nodes to infer. For example, AND-type Calendar tree has three cond-type nodes: Time, Confirm and Query.

Action-type node contains the actions to do when traversing to this node. Focus is found from the tree when the expression of a certain node is false, and the corresponding action is performed. Current focus isn’t finished until the expression becomes true.

![Figure 3. XML dialogue control tree.](image)

For calendar retrieval, inference starts at OR-type goal tree. Goal object is created first, and set to be Calendar. AND-type Calendar tree is visited, calendar_day, calendar_duration and calendar_hour slots are gotten and verified at Time node. Final confirmation is made at Confirm node if needs. Querying slots are sent to PIM server for SQL-based retrieval, and the calendar information at 4 o’clock this afternoon is returned. Goal object is then reset for next query at Query node.

![Figure 4. Multi-modal interaction for calendar task.](image)
For contact retrieval, AND-type Contact tree is visited, contact_name slot is copied from email_name slot, and the contact information of Bor-Shen’s email address is returned.

(a)

Figure 5. (a) Query result of email. System: There is one relevant email found from Bor-Shen ... (b) Query result of contact. System: Bor-Shen’s email address is bslin@itl.iii.org.tw.

3.3 Dynamic Presentation

Multi-media information is presented, and content format is dynamically changed according to current state. A virtual personal assistant converses to a user during the querying, with natural speech and abundant face expression, making the interface more human nature. Besides speech query, the form provides alternative way to be selected, and text box accepts user’s typing input, as showed in Figure 4(a). Speech input changes the state from unknown to known, and implicit verification is applied at next interaction, such as date slot described in Section 3.1. Finger touch changes the state from unknown to verified directly for its high reliability, such as time slot described in Section 3.1. Query result is gotten with synthesis speech and text output, as showed in Figure 4(b) and Figure 5.

4. INTEGRATION OF TECHNOLOGIES

ASR (Automatic Speech Recognition) converts speech into possible candidates in time spans. Considering the robustness of speech recognition, simply optimal recognition result is not enough for post-processing. Lattice or graph could contain much more information for NLP (natural language processing).

NLP (Natural Language Processing) parses and converts the ASR result into semantic slot. Grammar and concept N-gram language model are integrated to decide the optimal understanding result.

FD (Face Detection) applies digital image processing to detect whether a user appears in front of the system. Once face detected is gotten, it informs the system to start the service.

Talking Head synchronizes the synthesis speech with mouth action to simulate the real people speaking. It may be a famous news reporter or a cute cartoon character.

TTS, ASR, FD modules act as independent servers, communicating with service agent via MQ (message queue). Talking Head and DHTML are integrated to provide dynamic presentation, as shown in Figure 6.

(b)

Figure 6. Integration of technologies.

5. CONCLUSION

This paper presents a dialogue management approach providing dynamic and goal-oriented interaction. Several software technologies (ASR, TTS, DM, NLP, face detection and talking head) are integrated to construct an intelligent agent. It’s applied a PIM to provide goal-oriented management of personal email, contact, calendar, task and note, using natural speech, finger touch, or text input to interact with this service agent, in much more human nature and flexibility.

6. ACKNOWLEDGEMENTS

TTS module is provided by Applied Speech Technology corp. Talking head module is provided by BEXTech corp.

7. REFERENCES