

Utterance Planning in an Agent-based Dialogue System

Paul Thompson, Mark Stairmand and William Black

Department of Computation, UMIST, Manchester

Introduction

This poster describes the Response Planning and Generation components of the Athos (AKA Jaspis) framework implemented in the DUMAS project, which investigates adaptive multilingual interaction techniques to handle both spoken and text input and to provide coordinated linguistic responses.

AthosMail was developed in the DUMAS project as a multilingual speech based application enabling access to email by telephone. Also under development, AthosNews provides a speech interface to newspapers for visually impaired users, providing both browse and search functionality.

Jaspis Agents

Jaspis is an agent-based framework for adaptive speech applications

- Distributed, multilingual applications that adapt to the user and environment
- Small, stateless agents selected by Evaluators
 - Select the most appropriate agent to handle a dialogue situation
- Shared information store
 - Stores information relating to dialogue history, user modelling, etc.

Response Planner and Generator are Jaspis agents implemented in Java.

Response Planning

Jaspis agent PD Tree performs pragmatic analysis to produce goals

- anchoring, discourse modelling and interpretation of the user's goal
- goal represents how the system can fulfil the user's intention
- consists of a goal type, a query summary and a solution list

Response Planner selects and instantiates the appropriate utterance plan for the current dialogue state.

Based on templates encapsulating sets of conditions. Each 'response plan' has a corresponding template.

The goals supplied by PD Tree are unified with the templates to select the most appropriate response.

Example of a template for situation where an 'enumerate messages' response is appropriate:

```
[DG_OBJ(_C, 'MAIL', _T, _G, _S, _P)]
[DG_EVT(_E, _T, G, S, P) {0,}]
[DG_REL(E, _C, T, G, S, P) {0}]]
[DG_SOL(_V, G, T) {1,}]
```

Regexp-based notation, this signifies '0 or more'.

This is a nested clause, only evaluated in context of parent.

There must be a MAIL object referenced AND there can be 0 or more 'event' objects in the same proposition (BUT if there is an event, it must not be related to the MAIL object) AND there must be at least one solution in the same goal as the MAIL object.

Rule Evaluation

Evaluate next rule only if:

- there is more than one matching clause in dialogue model, unless the {0} case has been expressed AND
- any condition associated with the current rule evaluates to true.

Successful unification of all clauses in template means the utterance plan corresponding to the template is selected.

```
Database: DG_OBJ(10, 'MAIL', 4, 2, 1, 3)
          DG_EVT(17, 'READ', 7, 4, 2, 1, 3)
Query 1: DG_OBJ(_C, 'MAIL', _T, _G, _S, _P)
(next query uses instantiated variables)
Query 2: DG_EVT(_E, _T, 4, 2, 1, 3)
(next 'nested' query only done if previous returns a result)
Query 3: DG_REL(17, _C, 10, 4, 2, 1, 3)
```

Utterance Plans

An abstract plan that can be realised in different ways by the output generator.

Plans include a metalanguage for manipulation of unification results, . E.g.

- {C.from}: 'substitute the value of the from attribute for the first MAIL object bound in above unification.
- %C(): 'repeat the output in parentheses for each result bound to variable 'C'.

Uninstantiated Plan for 'enumerate messages' action:

```
%T (
[PRED text=enumerate]
[ARGO
[SET
[DESCRIPTION
[ATTVAL attribute='instanceOf' value='message']
[ATTVAL attribute='from' value='{C.from}]]
[MEMBERS %C(
[MEMBER
[ATTVAL attribute='instanceOf' value='message']
[ATTVAL attribute='index' value='${C.index$}'] )]] )
```

Instantiated utterance plan, produced from plan above, interpreting directives, based on variable values bound during unification:

```
<PRED>enumerate</PRED>
<ARGO> <SET>
<DESCRIPTION>
<ATTVAL attribute='instanceOf' value='message'></ATTVAL>
<ATTVAL attribute='from' value='Bob'></ATTVAL>
</DESCRIPTION>
<MEMBERS>
<MEMBER>
<ATTVAL attribute='instanceOf' value='message'></ATTVAL>
<ATTVAL attribute='index' value='2'></ATTVAL>
</MEMBER>
<MEMBER>
<ATTVAL attribute='instanceOf' value='message'></ATTVAL>
<ATTVAL attribute='index' value='5'></ATTVAL>
</MEMBER>
</MEMBERS>
</SET> </ARGO>
```

Surface Form Generation

Process utterance plans to select appropriate concatenations of pre-defined utterance segments.

User modelling component determines explicitness based on user experience and interaction.

There are [quantity] [message_description]. [message_details]. What would you like to do?

[quantity]: Number of 'member' elements
[message_description]: 'echo back' user selection criteria (cf. description)
[message_details]: info about retrieved messages using fields not specified in query, i.e. presents NEW information only.

Resulting surface form synthesized by text-to-speech engine

There are 2 messages from Bob. Message 1 dated 1st June 2003 with subject AthosMail and message 2 dated today with subject lunch. What would you like to do?

Acknowledgements

This work is carried out as part of the DUMAS project (Dynamic Universal Mobility for Adaptive Speech Interfaces), funded by the European Union IST Programme (contract IST-2000-29452).