Language Technology
http://cs.lth.se/edan20/
Chapter 17: Dialogue

Pierre Nugues

Pierre.Nugues@cs.lth.se
http://cs.lth.se/pierre_nugues/

October 10 and 17, 2019

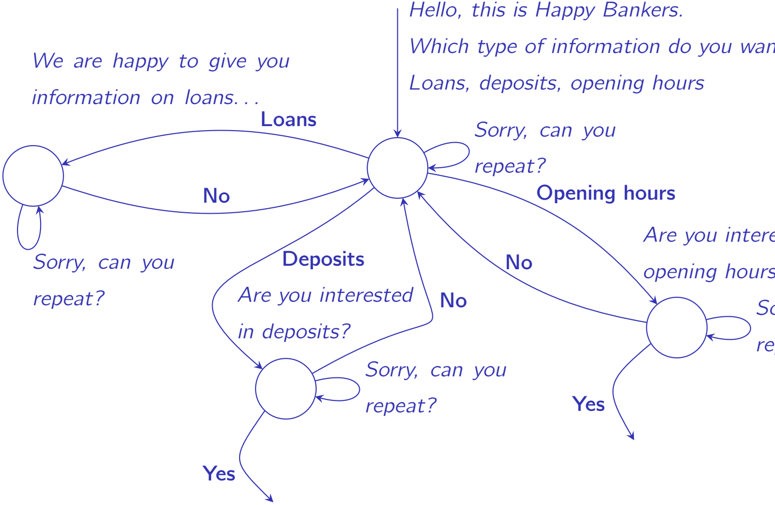
Dialogue

Interacting persons:

- Information can be missing
- Some words or constructions can be ambiguous,
- Errors in speech recognition.

Coreferences are central in a dialogue context.

Automata



Hello, this is Happy Bankers.

Which type of information do you want?

Loans, deposits, opening hours

Sorry, can you repeat?

Loans

We are happy to give you information on loans...

Sorry, can you repeat?

No

Deposits

Are you interested in deposits?

Sorry, can you repeat?

No

Yes

Opening hours

Are you interested in opening hours?

Sorry, can you repeat?

No

Yes

Dialogue Pairs

First member	Preferred second member	Dispreferred second member
Offer, Invitation	Acceptance	Refusal
Request	Compliance	Refusal
Assessment	Agreement	Disagreement
Question	Expected answer	Unexpected answer, no answer
Blame	Denial	Admission

More Elaborate Pairs

- *initiative interventions*, which open an exchange (I)
- *reaction interventions*, which are answers to initiatives (R)
- *evaluation interventions*, which assess exchanges and possibly close them (E)

Utt. no.	Turns	Utterances
1	S:	*Which type of information do you want: loans, deposits, opening hours?*
2	U:	*Loans*
3	S:	*We are happy to give you information on loans*

Pairs with Closed Questions

Utt. no.	Turns	Utterances	Tags
1	S:	*Which type of information do you want: loans, deposits, opening hours?*	I_1
2	U:	*Deposits*	R_1
3	S:	*Are you interested in deposits?*	I_1^2
4	U:	*Yes*	R_1^2

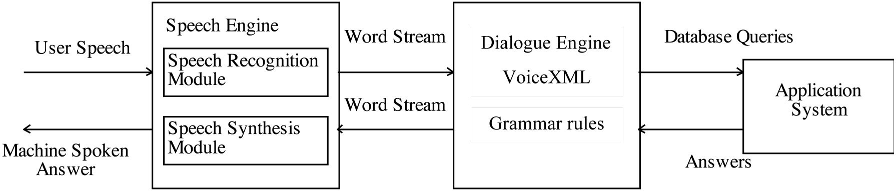
VoiceXML: A Language for Simple Dialogues

VoiceXML is a programming language to describe simple dialogues

It can process touch-tones, isolated words, and phrases with the help of a grammar

It uses external speech recognition and synthesis modules

It is frequently used in speech server applications



A VoiceXML Example

```
<?xml version="1.0" encoding="UTF-8"?>
<vxml xmlns="http://www.w3.org/2001/vxml" version="2.1"
  xml:base="http://localhost:8080/demo2/">
<form>
  <field name="process" type="bool">
    <prompt bargein="false">You have chosen the welding
      process. Do you want to continue?</prompt>
    <option>yes</option>
    <option>no</option>
    <filled>
      <prompt>You said <value expr="process"/></prompt>
    </filled>
  </field>
```

A VoiceXML Example

```
  <field name="calibration" type="calibration">
    <prompt bargein="false">Calibration of work piece...
    </prompt>
    <option>no</option>
    <option>manual</option>
    <option>automatic</option>
    <filled>
    <prompt>You said <value expr="calibration"/></prompt>
    </filled>
  </field>
...
</form>
</vxml>
```

Bühler's Sprechakte

- A representation (*Darstellung*) of objects and the state of affairs that is being described;
- An expression (*Ausdruck*) materializing the psychological state of mind of the speaker – the sender of the message;
- An appeal (*Appell*) corresponding to an effect on the hearer – the receiver of the message.

Speech Acts

Another classification due to Austin:

- *Locutionary* – saying something: syntactic structure, formal semantics content,
- *Illocutionary* – a conversational act: to inform, to suggest, to answer, to ask, . . .
- *Perlocutionary* effects: frighten, to worry, to convince, to persuade, . . .

Speech Acts Classes

- Assertives, such as stating, asserting, denying, informing;
- Directives, such as requesting, asking, urging, commanding, ordering;
- Commissives, such as promising, committing, threatening, consenting, refusing, offering;
- Declaratives, such as declaring the war, resigning, appointing, confirming, excommunicating. Declarative speech acts change states of affairs.
- Expressives, that are related to emotions or feelings such as apologizing, thanking, protesting, boasting, complimenting.

Links Between Syntax and Speech Acts

Classical grammar recognizes certain links between locutionary and illocutionary content:

Classical speech acts	**Syntactic forms**
Assertions, statements	Affirmatives or declaratives
Orders, commands	Imperatives
Questions	Interrogatives

Problem with Syntax

Syntactical form is sometimes misleading:

Can you open the door?	Question?
Have a good day!	Order?

A syntactical classification is too coarse to reflect the many needs of interaction analysis.

Searle's Conditions

For each act, four conditions. For example: assert

Conditions	**Values**
Propositional content	Any proposition P
Preparatory	① Speaker has evidence (reasons, etc.) for the truth of P ② It is not obvious to both Speaker and Hearer that Hearer knows (does not need to be reminded of, etc.) P
Sincerity	Speaker believes P
Essential	Counts as an undertaking to the effect that P represents an actual state of affairs

Searle's Conditions: Request, Order, Command

Conditions	**Values**
Propositional content	Future act *A* of Hearer
Preparatory	1. Hearer is able to do *A*. Speaker believes Hearer is able to do *A* 2. It is not obvious to both Speaker and Hearer that Hearer will do *A* in the normal course of events of his own accord 3. (For *order* and *command*) Speaker must be in a position of authority over Hearer
Sincerity	Speaker wants Hearer to do *A*
Essential	Counts as an attempt to get Hearer to do *A*

Helen of Troy

Speech acts are not new. Gorgias (487-380 BC) wrote:

> *The effect of speech upon the condition of the soul is comparable to the power of drugs over the nature of bodies. For just as different drugs dispel different secretions from the body, and some bring an end to disease and others to life, so also in the case of speeches, some distress, others delight, some cause fear, others make the hearers bold, and some drug and bewitch the soul with a kind of evil persuasion.*

Encomium of Helen (Trans. RK Sprague)

Speech Acts: Examples (Gazdar and Mellish, 1989)

Turns	Utterances	Illocutionary acts
A	I really think the automobile needs servicing	Statement
B	But we had done it recently	Challenge
A	No, not for two years. . .	Challenge
		Interruption
A	Incidentally did you hear that gas prices are about to double?	Concession

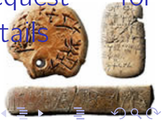
EVAR (Mast et al., 1994)

EVAR structures the dialogue process in a sequence of phases:

- A greeting by the system,
- A request from the user,
- A clarification consisting of a confirmation of data requested by the user and possible requests for details and specifications,
- An answer from the system, and finally
- A closing.

EVAR Dialogue

Turns	Dialogue	Phases
S:	*Guten Tag. Hier ist die automatische Intercity-Auskunft. Welche Information benötigen Sie?* 'Hello. This is the Automatic Travel Information System. What information do you need?'	Greeting
U:	*Ich möchte morgen nach Hamburg fahren.* 'Tomorrow I want to go to Hamburg.'	Request
S:	*Sie möchten morgen nach Hamburg fahren?* 'Tomorrow you want to go to Hamburg?'	Confirmation
U:	*Ja morgen.* 'Yes tomorrow.'	
S:	*Von wo möchten Sie abfahren?* 'Where do you want to start?'	Request for details
U:	*München* 'Munich'	



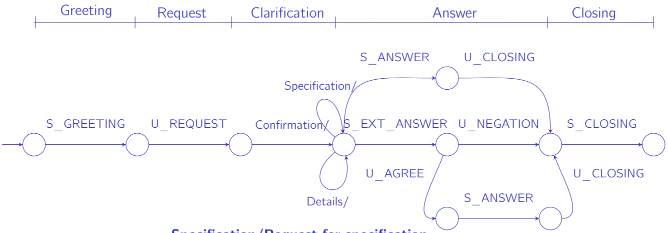
EVAR Dialogue

S:	*Morgen von München nach Hamburg* ‘Tomorrow from Munich to Hamburg’	
U:	*Ja* ‘Yes’	
S:	*Wann möchten Sie fahren?* ‘When do you want to leave?’	Request for specification
U:	*Morgen früh* ‘Early in the morning’	
S:	*Morgen früh von München nach Hamburg* ‘Tomorrow morning from Munich to Hamburg’	
U:	*Ja* ‘Yes’	

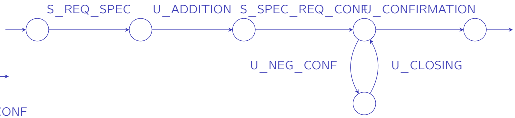
EVAR Dialogue

S:	*Ich habe mehere Verbindungen gefunden. Nach Hamburg HBF, fahren Sie ab München um 7 Uhr. Ankunft in Hamburg HBF um 15 Uhr, . . .* 'I found connections. From Munich departure 7, Arrival Hamburg 15, . . .'	Answer
U:	*Vielen Dank* 'Thank you'	
S:	*Möchten Sie weitere Auskünfte?* 'Do you want further information?'	Closing
U:	*Nein danke* 'No thank you'	
S:	*Auf wiederhören.* 'Have a nice trip.'	

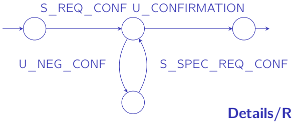
EVAR's Automata (Mast et al., 1994)



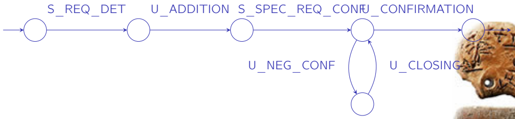
Specification/Request for specification



Confirmation



Details/Request for details



France Telecom CNET's Dialogue Acts (I)

Acts	**S/U**	**Descriptions**
Initiatives		
`request(P)`	`S`	Open question or request for the value of P
`yn_question(P, Val)`	`S`	Is value of `P` `Val`? Answer should be *yes* or *no*
`altern_question(P)`	`S`	Alternative question: *Vanilla or strawberry*?
`repeat(P)`	`S/U`	Repetition request
`inform(P)`	`S/U`	Inform of `P`
`recap(P)`	`S`	Recapitulation of solved problems

France Telecom CNET's Dialogue Acts (I)

Acts	**S/U**	**Descriptions**
Reactions		
`answer(P, Val)`	U	Gives a value `Val` on the request of `P`
`select(P, Val)`	U	Gives a value `Val` on an alternative question on `P`
`accept(P, Val)`	U	Accept or confirm the value `Val` of `P`
`reject(P, Val)`	U	Reject the value `Val` of `P`
Evaluations		
`impl_valid(P, Val)`	S	Implicit validation of confirmation of the value `Val` of `P`
`correct(P, Val)`	U	Gives a new value `Val` to `P`

A Dialogue Example

Turns	**Utterances**
S1	*Bonjour puis-je vous aider?* 'Hello, can I help you?'
U1	*Je veux une réservation de Londres à Paris* 'I want a reservation from London to Paris'
S2	*Londres Paris à quelle date voulez-vous voyager?* 'London Paris which date?'
U2	*Je veux voyager le 20 juin* 'I want to travel June 20'
S3	*Le 20 juin à quelle heure?* 'June 20 which time?'
U3	*À 17 heures 15* '5.15 pm'
S4	*17 heures 15?* '5.15 pm?'

A Dialogue Example

Turns	Utterances
S5	*Le vol BA-123 part à 17 heures 15 et arrive à Charles-de-Gaulle à 18 heures 39, est-ce que cela vous convient?* 'Flight BA-123 departure 5.15 pm and arrival 6.39 pm at Charles-de-Gaulle airport. Is that OK for you?'
U5	*Oui* 'Yes'
S6	*Quel est votre nom?* 'What is your name?'
U6	*Luc Martin*

A Dialogue Example

Turns	**Utterances**
S7	*Luc Martin une réservation pour Luc Martin sur le vol BA-123 arrivant à Charles-de-Gaulle à 18 heures 39* 'Luc Martin a reservation for Luc Martin on flight BA-123 arriving at Charles-de-Gaulle at 6.39 pm'
U7	*Oui* 'Yes'
S8	*Votre réservation a été confirmée au revoir* 'You reservation has been confirmed goodbye'
U8	*Au revoir* 'Goodbye'

Dialogue Decomposition

Exch.	Interventions	Recursive int.	Turns
	`Ii(s, [request])`		S1
E1	`Ir(u, [answer])`		U1
	`Ie(s, [impl_valid])`		S2a
	`Ii(s, [request])`		S2b
E2	`Ir(u, [answer])`		U2
	`Ie(s, [impl_valid])`		S3a
	`Ii(s, [request])`		S3b
E3	`Ir(u, [answer])`		U3
E3e		`Ie(s, [impl_valid])`	S4
		`Ir(u, [accept])`	U4

Dialogue Decomposition

Exch.	**Interventions**	**Recursive int.**	**Turns**
E4	`Ii(s, [recap, yn_question])`		S5a S5b
	`Ir(u, [accept])`		U5
E5	`Ii(s, [request])`		S6
	`Ir(u, [answer])`		U6
	`Ie(s, [impl_valid])`		S7a
E6	`Ii(s, [recap])`		S7b
	`Ir(u, [accept])`		U7
	`Ie(s, [impl_valid])`		S8

Speech Acts Recognition

The are based on:

- Cue words or phrases linked to specific speech acts
- Syntactic and semantic forms of the utterance
- Expectations to apply constraints on possible speech acts.

  These are based on transitions from a previous state to the current state of the dialogue: When the system asks a question, it expects an answer, a rejection or a failure, and it can discard other acts.
- Task modeling and goal satisfaction.

  It restrains possible user acts and parameter values according to the progress point where the user is in the dialogue.
- Recognition uses either machine-learning techniques to classify the speech acts or logical constraints.

Examples

Syntactic features	**Candidate speech acts**
Interrogative sentence	`yn_question`, `altern_question`, `request`
yes, right, all right, OK	`accept`, `impl_valid`
no, not at all	`reject`
Declarative sentence	`inform`, `impl_valid`
sorry, pardon, can you repeat	`repeat`
not X but Y, that's not X it's Y in fact.	`correct`

Beliefs and Intentions

Dialogue can be modeled in terms of agents with capacities:

- `wants(A, X)`, which means that agent `A` wants to do `X`,
- `can_do(A, X)`, which means that agent `A` can do `X`,
- `believes(A, X)`, which means that agent `A` believes `X`,
- `knows(A, X)`, which means that agent `A` knows `X`,

and acts: `informs(A, B, P)` (`A` informs `B` of `P`)
Preconditions and postconditions:

- Preconditions: `knows(A, P)`, `wants(A, inform(A, B, P))`
- Postconditions: `believes(B, P)`

An Operational Dialogue System: The SJ Train Information System

A service that answers questions on train times and fares in Sweden

Let the customers order tickets

Accessible by telephone: 0046 771-75-75-75

Based on a previous work done at Telia research

Paper reference: Johan Boye, Mats Wirén, Manny Rayner, Ian Lewin, David Carter, and Ralph Becket, “Language-Processing Strategies and Mixed-Initiative Dialogues”, *IJCAI-99 Workshop on Knowledge and Reasoning in Practical Dialogue Systems*, July 1999

Corpus Collection

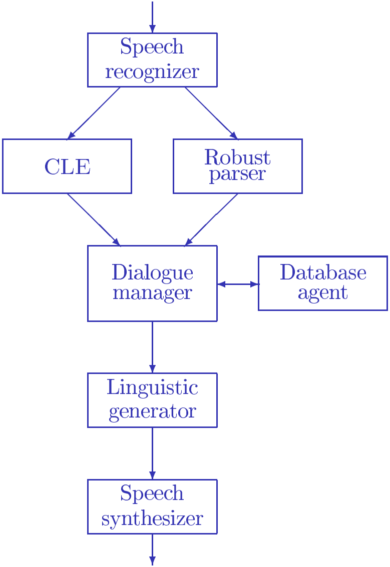
The authors started the development with a corpus collection

Used the wizard-of-oz technique: ˜130 dialogues and 50 subjects:

- Gives the vocabulary
- Gives the basic structure of a dialogue: a specification phase followed by a negotiation phase

Considerable variation amongst the customers.

Architecture


Speech recognizer

CLE

Robust parser

Dialogue manager

Database agent

Linguistic generator

Speech synthesizer

Dialogue Example

> *Hej jag beställer en flygbiljett den artonde i sjätte tisdag från Stockholm till Sundsvall.*

recognized as

> *vad hej jag beställer jag vill jag den artonde i sjätte i jag mmm då Stockholm till Sundsvall.*

Formal Representation

The system uses a “flat utterance description”: a predicate whose argument is a list of parameter/value pairs

Four possible predicates corresponding to the customers’ requests:

Act	**Description**
`yn`	Are there objects with property `P`?
`wh`	Find `X` with property `P`
`wh_agg`	Find the maximal/minimal `X` with property `P`
`yn_agg`	Does the maximal/minimal `X` with property `P` also have property `P’`

Examples

The utterance

I want to arrive in Stockholm before 6 pm

is interpreted as

Find flights arriving Stockholm before 6 pm.

It is translated as:

```
wh(X, [slot(trip, trip_id, X),
    slot(trip, trip_mode, plane),
    slot(trip, to_city, Stockholm),
    slot(trip, arr_time, T),
    exec(before(T, 1800))])
```

Examples

The utterance

Is that a direct flight?

is represented by:

```
yn([slot(trip, trip_mode, plane),
    slot(trip, stops, 0),
    slot(trip, trip_id, X),
    ref(X, det(def, sing))])
```

Examples of the two remaining acts:

`wh_agg`: *I want the first flight to Stockholm?*
Which is the cheapest ticket?

`yn_agg`: *Is that the first flight?*

Parsing

Dual system:

- A robust parser – a partial parser – identifies phrases and keywords (DCG). The phrases are used to fill in the slots.
- The core language engine from SRI directly outputs FUDs using compositional rules

The robust parser is overall more efficient than the core language engine. See discussion in the paper.

Dialogue Acts

An example of dialogue between a user (U) and the system (S).

	Utterance	Act
U	*I want to go from Gothenburg to Stockholm on Friday*	`user:constraint`
S	*At what time do you want to leave?*	`system:ask-for-constraint`
U	*In the morning*	`user:constraint`
S	*There is a train at 5:30 am arriving at 9:45 am*	`system:suggestion`
U	*Is that a direct train?*	`user:ask-for-info`
S	*Yes*	`system:answer-with-info`
U	*Is there a later train?*	`user:ask-for-suggestion`
S	*There is a train at 6:06 arriving at 9:15*	`system:suggestion`
U	*Fine, I'll take that one*	`user:accept`

Dialogue Act Identification

In total, 12 different dialogue acts identified using the dialogue state and rules:

- The existence of suitable contexts. For example, an utterance cannot be classified as a `user:accept` unless the system has proposed some train(s) and or flight(s) that the user can accept
- The difference between the propositional contents of the utterance and that of the context. For instance, if these two are inconsistent, the utterance cannot be classified as a `user:accept`; if they are consistent, it is unlikely that the utterance should be classified as a `user:ask-for-suggestion`.
- The presence of keywords in the utterance. For example, if the utterance contains "accept words" like *yes*, *ok*, etc., the `user:accept` score is increased