

Modelling of Natural Dialogue Systems

Challenges and Approaches

University of Ulster, Belfast, 18 September 2013



Markus M. Berg
University of Kiel
University of Wismar

<http://mmberg.net>

Context: Spoken Dialogue Systems

“A spoken dialogue system enables a human user to access information and services that are available on a computer or over the Internet using spoken language as the medium of interaction”

[Jokinen and McTear]

- Control Systems (Smart Rooms)
- Booking and Information Systems
 - Interactive Voice Response Systems

Example Dialogue (I)

- Welcome to the Flight Information System.
Where do you want to start?
- *Belfast*
- Where do you want to go?
- *San Francisco*
- When do you want to leave?
- *01/10/2013*
- When do you want to come back?
- *14/10/2013*

Example Dialogue (II)

- Welcome to the Flight Information System.
Where do you want to start?
 - *I'd like to go from Belfast to San Francisco*
 - Ok, you start in San Francisco. Where do you want to go?
 - *No, Belfast!*
 - When do you want to leave from San Francisco?
 - *I want to leave Belfast next Monday!*
 - When do you want to come back?
 - ...
- ▶ Over-informative answers, corrections, colloquial utterances

Motivation (I)

- SDS regarded as being tedious and inflexible
- Major problems [Bringert]
 - Not usable enough ▶
 - Not natural enough ▶
 - Not cheap enough ▶
 - (no suitable tool or model ?!)*
- 71% wish for human-like communication with robots [Dautenhahn]
- Current commercial dialogue systems show only limited capabilities with regard to natural dialogue [Pfleger]

Motivation (II)

- True mixed-initiative dialogue management is only available in research prototypes [Pfleger]
- Several research prototypes are created from scratch
 - *eOhr* (SDS, Smart Room)
 - *TravelConsult* (Chatbot, Virtual Travel Agency)
 - *Virtual Reception Desk* (mobile SDS, Trip Information)
- A generic dialogue model and corresponding dialogue engine that can be reused across different projects would be beneficial

Main Objectives

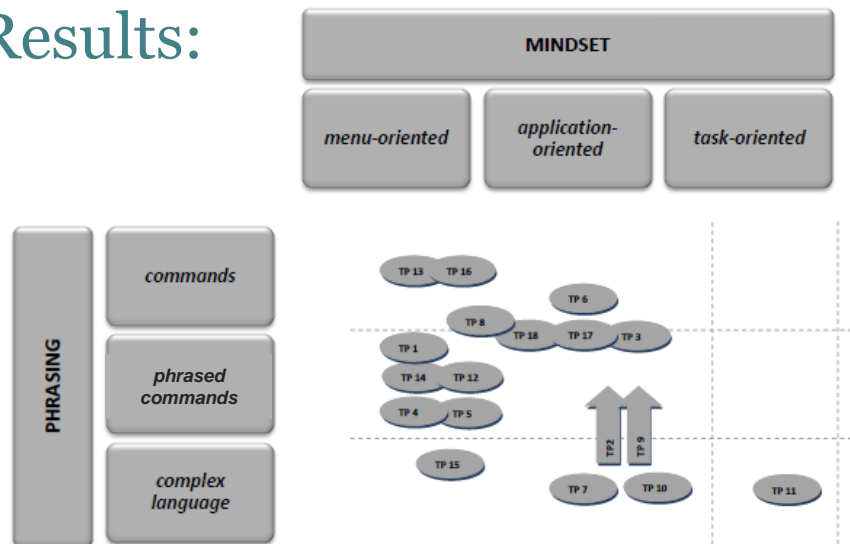
- What problems exist in the interaction between man and machine and how do humans prefer to communicate with automated systems?
- How can a natural dialogue be described so that it is automatically processable by a dialogue engine?

Aspects

- Analysis of human communication with speech interfaces
 - What makes a dialogue natural?
- Classification and modelling of questions and answers as well as their effect on the back end
- Language generation and linguistic style variation
- Development of a dialogue model that can be
 - automatically processed by a dialogue engine
 - created with the help of a dialogue development environment

User Study

- Wizard of Oz experiment
 - Human operator simulates SDS (TTS)
 - 18 participants
 - Goal: find out *how* people interact with SDS
 - Results:



„Inbox!“ / „File, New“ / „Logout“

vs.

„Show new emails folder“

vs.

„Could you please show me my new emails“

Survey

- user expectations regarding the style of a dialogue system
- 194 participants
- Among other things: preferred dialogue

Dialog A:

Hello, this is Lena. I am an electronic travel agent. How may I help you?

In August I'd like to go to Spain with my wife. Which travels can you offer?

How long do you want to go away?

Two weeks.

...

Dialog B:

Welcome to your electronic travel agency system. You now have the following choices: trip information, book reserved trip, or cancel trip.

trip information

Where do you want to go to?

Spain

With how many persons?

Two

When do you want to depart?

in August

For how long?

Two weeks

...



Conclusions

- 71% prefer a human-like communication with machines
- current systems have too many options that confuse the user
 - often the user does not know which category to choose
- 80% want to describe their problem actively instead of answering lots of questions
- only 16% think politeness in dialogue systems is unnecessary
- in control situations 61% use short commands
- when the task gets more complicated users tend to use a more complex and complete formulation
- many people are mentally conditioned when it comes to operating machines: they try to imagine a GUI
- many people use short commands because they do not expect systems to understand full sentences
- no clear relation between age, profession or gender and preferred style

Utterances and Their Effect on the System

- **Problem: Misinterpretations**

S: Please tell me your destination!

U: Is there a direct connection to London City?

S-a: Ok, your destination is London City. Where do you want to start?

S-b: No!

U-b: Ok, then I'd like to go to Heathrow.

- **Dialogue system: three basic functions**

- the user wants the system to realise a request (a command)
- the user wants to retrieve information from the system (a question)
- the user gives information in order to enable the system to provide him with information (an answer)

System-oriented Dialogue Acts

- Information-seeking
- Information-providing
- Action-requesting

- Delimit from form and style
- Focus on the effect on the system
- remove ambiguity from user utterances without the necessity for a deep understanding
- *“Show me trips to London“ == “Do you have trips to London?“*

- help to bridge the gap between current industry applications and research prototypes

Recognition of Dialogue Acts

- “**Would you please shut down the projector?**”
→ *do* (action requesting)
- “**Would you please tell me when the next bus leaves?**”
→ *getInfo* (information seeking)
- No difference in form, but different back-end action

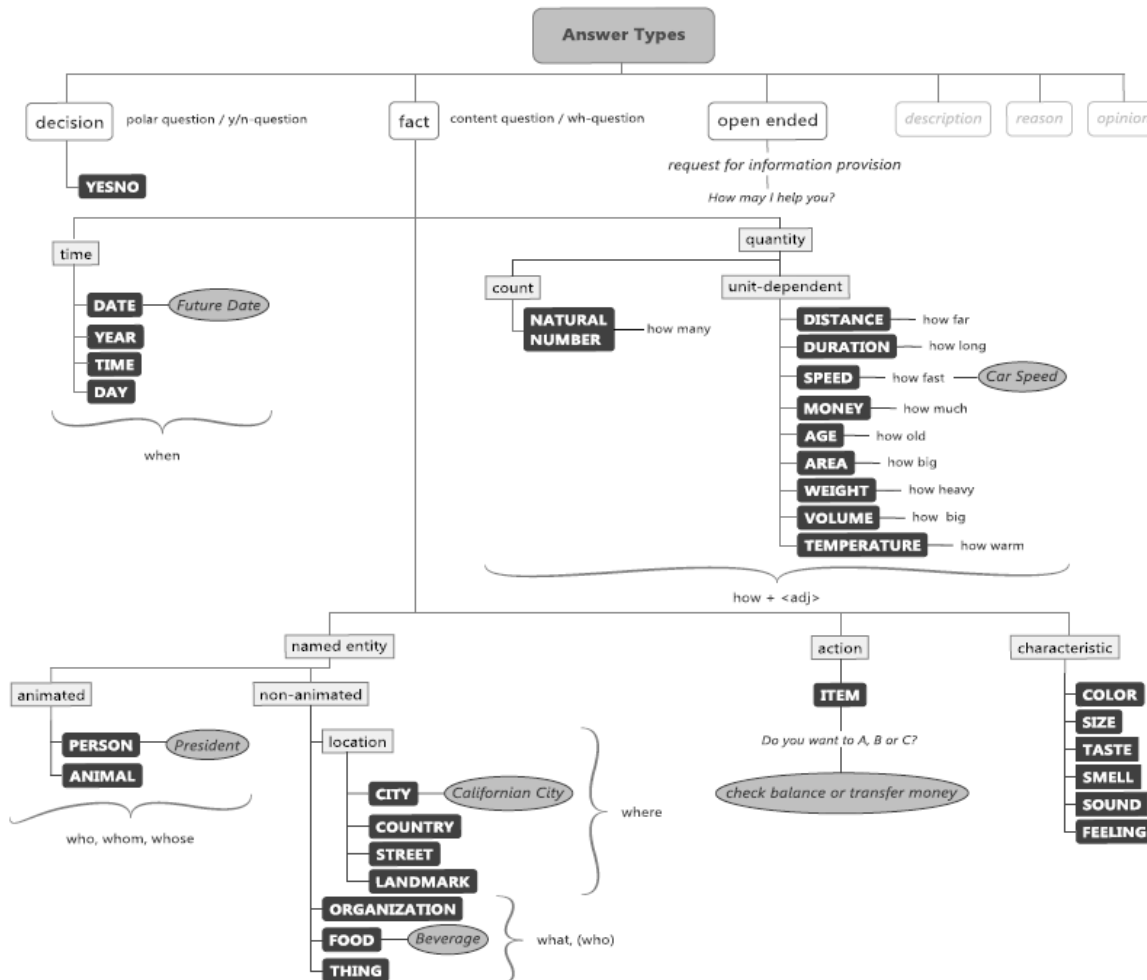
<i>class</i>	<i>instance</i>	inf. seeking	inf. prov.	act. req.
mood	indicative	×	×	×
	conditional	×		×
	imperative	×		×
	interrogative	×		×
verb category (<i>cue verbs</i>)	information gathering	×		
	action executing			×
question word	Wh	×		

- Boolean features: *cue verbs* (*shut, close, switch* vs *tell, know*)
- + wh-word, conditional, interrogative
- Maximum Entropy Classifier (without cue verbs only 38% precision)

Categorisation of Questions and Their Answers

- We know about the general intention of the user
- Now focus on information-seeking acts and their replies
 - generate a system-question of a specific type
 - process the user's answer to this question
- Classification of questions:
 - by question words: ambiguous (e.g. “*what*” does not refer to a specific type)
 - by answer type [Srihari and Li]: weight, area, temporal,...

Answer Type Hierarchy



Abstract Question Description

- Type
 - Answer Type
 - Reference Type
 - Cardinality
- Form
 - Question Type
 - Language
 - Surface Modifier
- Context
 - Purpose
 - Role
 - pragmatic aspect
 - reference




What the question asks for

▶ `fact.namedEntity.nonAnimated.location.city`

e.g.

- *Where do you want to go to?*
- *What's your destination?*
- *Do you already have a destination in mind?*
- *I need to know where you want to go to.*
- *Tell me about your travel plans!*

Abstract Question Description

- Type
 - Answer Type
 - Reference Type
 - Cardinality
- Form 
 - Question Type
 - Language
 - Surface Modifier
- Context
 - Purpose
 - Role
 - pragmatic aspect
 - reference

How the question is asked

- Question Type:

Question (Y/N): *Do you want to go?*

Wh-Question: *When do you want to go?*

Wh-Request: *[Please] Tell me / specify when you want to go!*

NP-Request: *[Please] Tell me your departure time [please]!*

C-Wh-Question: *Can/Could you [please] tell me when you want to go?*


C-NP-Question: *Can/Could you [please] tell me your departure time?*

Command (NP): *Your departure time?*

Command (N): *Departure time...?*

- Language
- Surface Modifier (disjunctive, politeness, formality)

Abstract Question Description

- Type
 - Answer Type
 - Reference Type
 - Cardinality
- Form
 - Question Type
 - Language
 - Surface Modifier
- Context 
 - Purpose
 - Role
 - pragmatic aspect
 - reference

What the question is about

- Described by the *role* (pragmatic aspect, reference)

“When do you want to leave?” →
type: fact.time.date{x > today}
role: start of trip

“When do you want to come back?” →
type: fact.time.date{x > today}
role: end of trip

“Where do you want to go?” →
type: fact.namedEntity.nonAnimated.location.city
role: end of trip

“When were you born?” →
type: fact.time.date
role: begin of life/person

- Important for language generation
- (but also for NLU)

Concept-to-Text

- Generate utterances based on abstract descriptions → AQD
- *“ask for the departure date in an informal way and don't make special use of politeness”*
- *Politeness*
 - Structure, e.g. interrogative style (CCG Grammar, Request/Question)
 - Subjunctive, please
- *Formality*
 - Structure (CCG grammar, NP/VP)
 - Lexicalisation (Ontology)
 - T/V-distinction
- Together represented as a logical form

```
s { stype=c-question } :  
  @w0( can ^  
    <mood>subj ^  
    <actor>(w1 ^ pron ^  
      <num>sg ^  
      <pers>2nd) ^  
    <theme>(w2 ^ tell ^  
      <patient>(w3 ^ pron ^  
        <num>sg ^  
        <pers>1st) ^  
      <theme>(w4 ^ when ^  
        <prop>(w6 ^ want ^  
          <agent>(w5 ^ pron ^  
            <num>sg ^  
            <pers>2nd) ^  
          <theme>(w8 ^ leave ^  
            <agent>x1))))))
```

“Could you please tell me when you want to leave?”

Concept-to-Text

- Generation based on:
 - Part of speech
 - Domain
 - Meaning $\mu = \{\text{dimension, pragmatic aspect, reference}\}$

go (v) \in Travel

$\gamma = \text{temporal} \wedge \text{begin} \vee \text{local} \wedge \text{end}$

$\rho = \text{trip}$

▷ *When do you want to go?*

▷ *Where do you want to go?*

“go” is a verb in the travel domain that can be used in a temporal context to describe the beginning of a trip or in a local context to describe the end of a trip.

AQD: Concept-to-Text

- Integrate model into AQD
- *“Could you tell me when you want to leave?”*
 - Type:
 - Answer Type & Constraint: `fact.temporal.date{x>today}`
 - Form
 - Question Type: `C-Question`
 - Language: `English`
 - Surface Modifier: `politeness=4, formality=2`
 - Context
 - Purpose: `gather information`
 - Role
 - * Pragmatic Aspect: `start`
 - * Reference: `trip`

API & Results

```
questions.add(new Meaning("fact.temporal.date", "begin", "trip"));
questions.add(new Meaning("fact.temporal.date", "end", "trip"));
questions.add(new Meaning("fact.location", "begin", "trip"));
questions.add(new Meaning("fact.location", "end", "trip"));
questions.add(new Meaning("decision", "possession", "customer_card"));
```

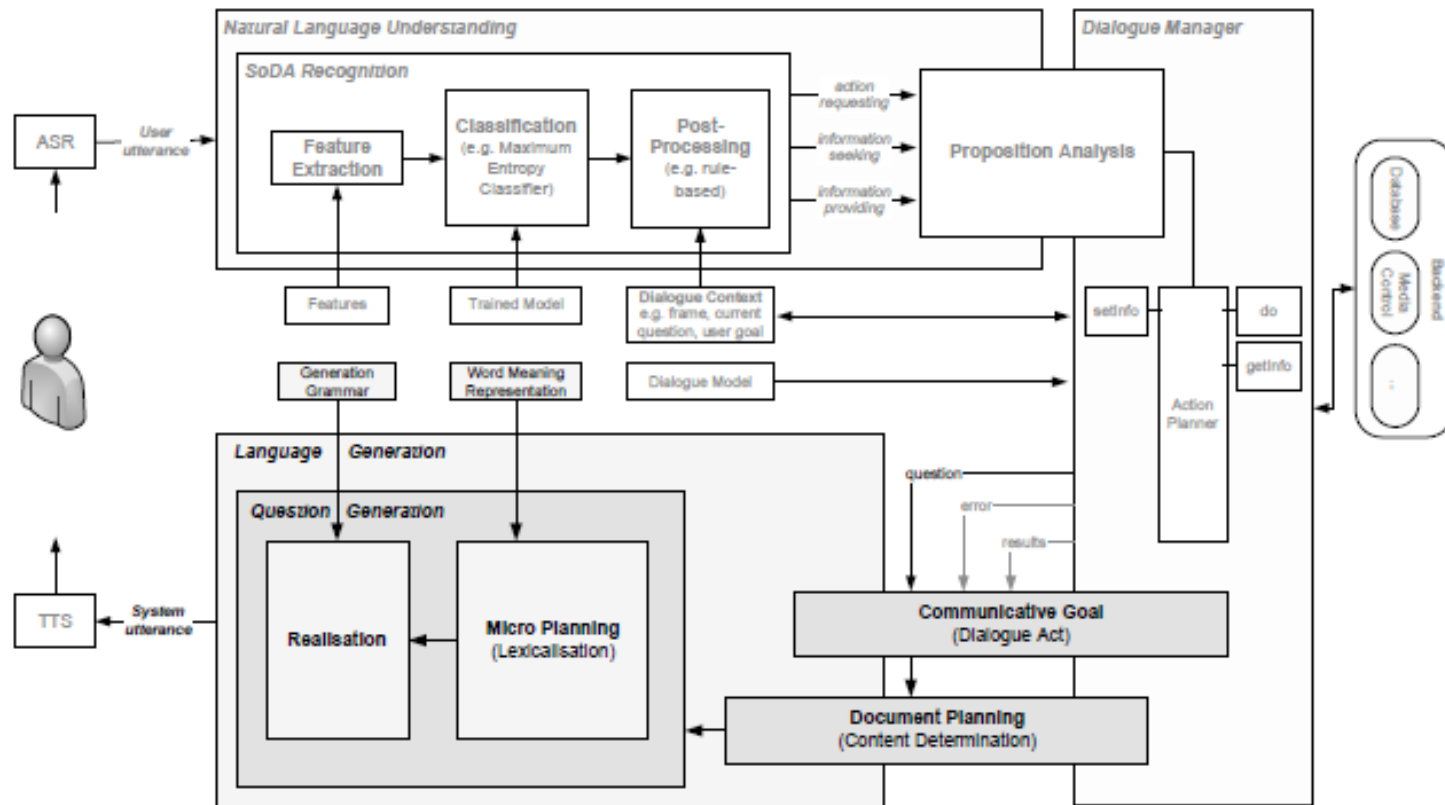
S: Departure date please!
U: ...
S: Now tell me your return date!
U: ...
S: Tell me your departure city!
U: ...
S: And the destination please!
U: ...
S: Do you have a customer card?
U: ...

Formality =2, politeness=1

S: When do you want to travel?
U: ...
S: Can you now tell me when you want to return?
U: ...
S: Please tell me your departure city!
U: ...
S: And where do you want to go?
U: ...
S: Do you have a customer card?
U: ...

Formality =2, politeness=4

Dialogue System Overview



Dialogue Model: Goals

- separate the dialogue manager/system from the dialogue specification
- focus on the definition of the information units that are required to full a task by combining the description of questions and their possible answers
- choose from a set of dialogue strategies instead of defining them repeatedly
- use language generation methods in order to abstract from language and style and make the creation of adaptive systems possible
- use predefined natural language understanding modules that can be selected by an abstract question description

Dialogue Model

- Dialogues and Tasks

- Natural communication includes subdialogues

S: Hello, how may I help you?

U: I'd like to book a trip.

S: Where do you want to start?

U: In Hamburg.

S: And where do you want to go?

U: **Um, can you tell me something about Paris?**

S: Paris is the capital of France and is called the city of love. It offers a variety of interesting sights, e.g. ...

S: Please tell me where you want to go!

U: **Well, how is the weather in Paris?**

S: 22°C and sunny.

S: Where do you want to go?

U: Ok, Paris then.

S: Can you now tell me when you want to leave?

U: ...

Information Transfer Objects

- Strong relationship between question and answer
- Answers can only be interpreted in the context of the question
- ITO
 - name
 - AQD
 - alternative utterance
 - ...

```
[ name :      getDepartureCity  
  group :      1  
  index :      0  
  fallback :   Where do you want to start?  
  useLG :      yes  
  AQD type :   fact.named_entity.non_animated.location.city  
  AQD context : begin,trip  
  AQD form :   n/a ]
```

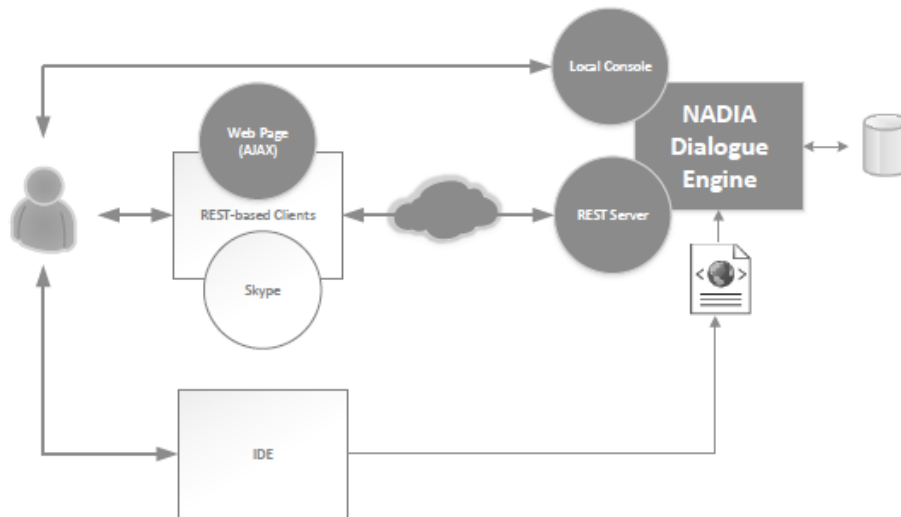
XML Model

- Dialogue > Task > Selector, ITO, Action
- Selector identifies responsible Task
- Action is executed on the back end
 - Groovy
 - Java (Class loader)
 - HTTP (REST) + XML/XPATH

Task: getWeatherInformation
Selector > bagOfWordsTaskSelector weather forecast temperature
ITO: getWeatherCity <i>answerType: fact.named_entity.non_animated.location.city</i> <i>fallback_question: For which city do you want to know the weather? group: 0 index: 0 required: false useLG: false</i>
Action > httpAction <ul style="list-style-type: none">• <i>returnAnswer: true</i>• <i>utteranceTemplate: The temperature in %getWeatherCity is #result degrees.</i>• <i>method: get</i>• <i>params: q=%getWeatherCity&mode=xml&units=metric</i>• <i>url: http://api.openweathermap.org/data/2.5/weather</i>• <i>xpath: /current/temperature/@value</i>

Natural Dialogue System (Nadia)

- Dialogue Engine that processes the model
- REST based, war/jar (embedded Jetty)
- Several user interfaces possible
- Web UI: Google TTS/ASR



```
curl -k --data "userUtterance=I+want+to+book+a+trip"
https://localhost:8080/nadia/engine/dialog/d1
```

Natural Dialogue System

System: How may I help you?

You:

<https://localhost:8080/nadia/engine/dialog/d1>
[show context](#) | [show dialogue](#)

© 2013 by [Markus M. Berg](#) | Project on [Github](#)
University of Wismar, University of Kiel, University of Edinburgh

Nadia Processing

- Identify SODA
 - if information providing
 - check if the utterance could be a correction
 - check if the utterance is an answer to a different question from the same task
 - check if a different task may be responsible and if so interpret the utterance in that task
 - repeat the question if an interpretation is impossible
- Get task, get responsible question
- Perform NLU
- Get next question (until frame filled)
- Execute action
- Follow-up question / stacked task / open-ended question

Dialogue Acts

```
Dialog: dummy2
start_task_name: start global_language: en global_politeness: 4 global_formality: 4 useSODA: true
Task: start
Selector > bagOfWordsTaskSelector
```

System: **How may I help you?**

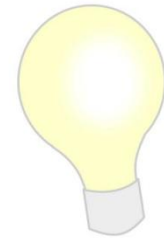
You: I want to know if the light is switched on

Start recognition Submit

System: **The light is switched off.** How may I help you?

You: I want to know if the light is switched on

Start recognition Submit



System: **The light is switched off. How may I help you?**

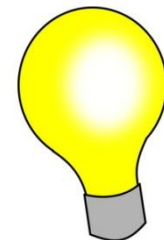
You: Switch the light on

Start recognition Submit

System: **The light has been switched on.** How may I help you?

You: switch the light on

Start recognition Submit



```
INFO (MaximumEntropyModel): Utterance: 'I want to know if the light is switched on' resulted in: seek[0,5002] action[0,4905] prov[0,0093] -> seek
```

```
INFO (MaximumEntropyModel): Utterance: 'switch the light on' resulted in: seek[0,0401] action[0,9420] prov[0,0179] -> action
```

Dialogue Behaviour

- Mixed Initiative
 - Switch Tasks
 - Support subdialogues
 - Over Answering
 - Provide more information than is asked for (but still answer current question)
 - Different Question
 - Ignore current question but answer another (unanswered) one
 - Correction
 - Replace an answer with a new one

Dialogue Extract

- How may I help you?
- *I'd like to get price information about a flight.*
- Where do you want to depart?
- *Hamburg.*
- Can you please tell me where you want to go?
- *Can you tell me something about Edinburgh?*
- Edinburgh is the capital city of Scotland, situated on the southern shore of the Firth of Forth. With a population of 482,640 [...]. Where do you want to go?
- *How is the weather in Edinburgh then?*
- 16° C. Can you tell me where you want to go?
- *Okay, I want to go to Edinburgh.*
- When do you want to depart?
- ...

Benefits

- User adaptive formulations: language generation
 - No need to specify static strings
- Reusable NLU modules
 - Addressed by type hierarchy
- No need to describe dialogue behaviour / flow
- Real subdialogues and switching between tasks
- Model could be run by different engines
- Separation of application and dialogue (dialogue as a service)
- Platform independent through the use of XML and REST
- Graphical IDE is currently being developed

Thank you very much for your
attention!

Questions?



© freedigitalphotos.net / Master isolated images