Role-plays for CALL: System Architecture and Resources

Sabrina Wilske & Magdalena Wolska
sw@coli.uni-sb.de
Saarland University
ICL, Villach, September 2008
Motivation

- Practicing communicative skills is an important aspect in foreign language learning (→ Communicative Approach)
- One technique: role-plays - interactive activity grounded in a well-defined authentic real-world situation
  - Learner has a task or a problem to solve
    - Basic social interactions (greetings, introductions)
    - Hotel booking, ordering in a restaurant, shopping, etc.
- What if you don't have anyone around to conduct role-plays with?
  → Use a machine!
Goals

• Dialogue system for roleplay-like dialogues

• Use insights from dialogue systems technology to build an architecture for language learning dialogues
  whose resources (e.g. linguistic) can be authored by non-technical developers (e.g. language teachers)

• long term objective: spoken interaction
  present prototype: type-written
Goals - Example Target Dialogues

W1: Good evening, Sir.

G1: Good evening.

W2: What would you like as a starter?

G2: Can you bring me a green salad, please?

W3a: Sure.

W3b: What would you like for the main course?

G3: I’d like the Marinated Flank Steak.

W4: How would you like your steak: rare, medium, or well done?

G4: Well done, please.

W5: Alright.

W1: Good Evening, are you ready to order?

G1: Sure, what would you recommend for a starter?

W2: Our Caesar Salad is very good.

G2: I’ll have the Caesar Salad then.

W3a: Okay.

W3b: And what would you like for the main course?

G3: Could I have the Beef Stroganov?

W4: Sure. Would you like potatoes or rice with that?

G4: I’ll have the potatoes please.

W5: Okay..
Goals: Dialogue Activity

• Various scenarios (in a hotel, in a restaurant, ...)

• Different configurations of a particular scenario
  • Different instantiations of the scenario setup (e.g. different restaurants, different menus)
  • Following different paths in a dialogue (e.g. order a starter or not)
  • Using different realizations of communicative goals
Outline

- System Architecture
  - Overview on Components
    - Language Model
    - Dialogue Model
    - Domain Model
  - Interaction between Components
- Illustrating Example
- Summary
- Problems, Further Work
In order to realize these dialogues, we need the following components:

- **Specification of the dialogue flow**: \(\rightarrow\) Dialogue Model
  alternative dialogue flows

- **Specification of the domain knowledge**: \(\rightarrow\) Domain Model
  different scenarios and different configuration of the scenario

- **Relevant communicative acts and utterances in the given scenario**: \(\rightarrow\) Language Model
  different linguistic realizations of communicative acts

- **Authoring tasks divided**:
  - Language Model/Dialogue Model \(\rightarrow\) adqjdjh#hdfkhuw
  - Domain Model \(\rightarrow\) adqjdjh#hdfkhuwB#h{shuw#q#grp dl# rhgdqj,
System Architecture: Components

Dialogue Model

Dialogue flow

Domain knowledge

Utterances and communicative acts

Language Model

Domain Model

\[ ut-1 := \ldots \ldots \ldots \]
\[ ut-2 := \ldots \text{slot-x} \ldots \ldots \]
\[ \vdots \]
\[ ut-n := \ldots \text{slot-x} \ldots \text{slot-y} \ldots \]
System Architecture: Language Model

Dialogue Model

Language Model

Domain Model

utterances and communicative acts
Language Model

- Mapping between utterance types (communicative acts) and set of strings that can realize them
- Implemented as context-free grammar
  - Templates with slots for ‘fillers’ from the domain model

\[
\text{REQUEST}(X) := \text{I would like } X \text{ please.}
\]

Generating system utterances:
random utterance from set of utterances for given type

Understanding user utterances:
matching semantic tags

\[
\text{REQUEST}(X) := \text{I would like } X\{X\} \text{ please } \{\text{request}\}
\]
System Architecture: Dialogue Model

**Dialogue Model**

- ut-1
- ut-2
- ut-4
- ut-6
- ut-7
- ut-8

**dialogue flow**

**Language Model**

- ut-1 := ... ...
- ut-2 := ... slot-x ...
- ...
- ut-n := ... slot-x ... slot-y...

**Domain Model**
Dialogue Model

- State-based representation of sequences of utterances
- Depending on requirements of the scenario: linear, branching, looping
- Implementation: dialogue management tool DialogOS (extendable platform allows to connect arbitrary components and exchange information with them)
System Architecture: Domain Model

Dialogue Model

Language Model

Domain Model

domain knowledge
Domain Model & Microworld

- Encodes domain knowledge by specifying relevant concepts, entities, and relations
- Also knowledge relevant to dialogue flow (at the moment)
- Formulated as an ontology (at the moment)

**Microworld**: Provides the context of learner’s task (entities they can refer to)
- Specific instantiation of a subset of the domain model
System Architecture: Components

**Dialogue Model**
- ut-1
- ut-2
- ut-3
- ut-4
- ut-5
- ut-6
- ut-7
- ut-8

**Dialogue flow**

**utterances and communicative acts**

**Language Model**
- ut-1 := ...
- ut-2 := ... slot-x ...
- ...
- ut-n := ... slot-x ...slot-y...

**Domain Model**

**domain knowledge**

**instance**

**Microworld**
- Menu-Header x
  - item 1
  - item 2
- Menu-Header y
  - item 3
  - item 4
Outline

• Motivation & Goal definition
• System Architecture
  • Overview on Components
    • Language Model
    • Dialogue Model
    • Domain Model
  • Interaction between Components
• Illustrating Example
• Summary
• Problems, Further Work
Interaction between modules

- Building blocks of dialogue model are utterance types
- Separation allows independent development and relatively easy adaptation to other languages
Interaction between modules

- Utterance template abstracts away from specific content words by specifying slots to be filled with appropriate items from the microworld.

- Allows automatic modification of the language model according to a given microworld.
Interaction between modules

- Sequence of utterances in overall dialogue structure is informed by the domain model.
- While executing the dialogue the dialogue manager refers directly to concepts of domain model.
- Features of concepts can trigger subdialogues.
Example: Domain Model & Microworld

Menu

Starters
- Salads
- Green Salad
- Mixed Salad

Soups
- Corn Soup
- Chicken Soup

Main Courses
- Lasagne...
- Fish...
- Steak
- Sirloin Steak
- Marinated Flank Steak
Example: Dialogue Model

GREETING

GREETING

for each course\(_i\) \in\) Courses

ASK.for(course\(_i\))

REQUEST(x\in course\(_i\))

x has undiscussed property slot

ACKNOWLEDGE

x has no undiscussed property slot

REQUEST(x,p\in list\(_px\))

ASK.how(x,list\(_px\))

PREP method

rare, medium, well done
Example: Dialogue Model – Instance 1

Good evening, Sir.

Good evening.

What would you like as a starter?

Can you bring me a green salad, please?

Sure.
Example: Dialogue Model – Instance 2

Good evening, Sir.

Good evening.

What would you like for the main course?

I’d like the Marinated Flank Steak, please.

How would you like your steak: rare, medium, or well done?

Alright.

Well done, please.
Summary

• Presented an architecture for executing language learning dialogues

• Key: separate domain knowledge from dialogue structure and language model → with view to independent development of system resources

• Allows to conduct dialogues in different scenarios and different variants of one scenario

• Implemented for the Restaurant and Shopping scenario
Problems / Further work

• Authoring domain model requires expert background in knowledge representation (ontology modeling)

→ Other (easier) ways to represent domain knowledge (targeting non-technical authors)

• Further work on the format of the language model (with respect to ease of authoring for language teachers)
  • Is there a standard for encoding language learning relevant information at the utterance level?

• Include other parameters: proficiency level, register, social context

• Evaluation of System:
  • Usability (for Learners and Authors)
  • Learning gains
Thank you.