



TECHNISCHE
UNIVERSITÄT
WIEN
Vienna University of Technology

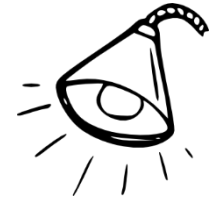


Institut für
Computertechnik
Institute of
Computer Technology

Towards Semi-Automatic Generation of Vocal Speech User Interfaces from Interaction Models

Dominik Ertl and David Raneburger

Agenda



- Motivation
- Background – Discourse-based Communication Model
- Exemplary Semi-Automatic Generation Approach for Command-based Speech In- and Output
 - Speech Metamodel
 - Speech UI Generation Approach
- Discussion
- Conclusion

Motivation

- UI Generation typically focuses on WIMP-UI Generation -> other modalities for more „natural“ communicative interaction
- Variety of different speech in- and output toolkits
- Reduce UI development effort through semi-automated generation
- Command recognition easier to implement than natural language recognition

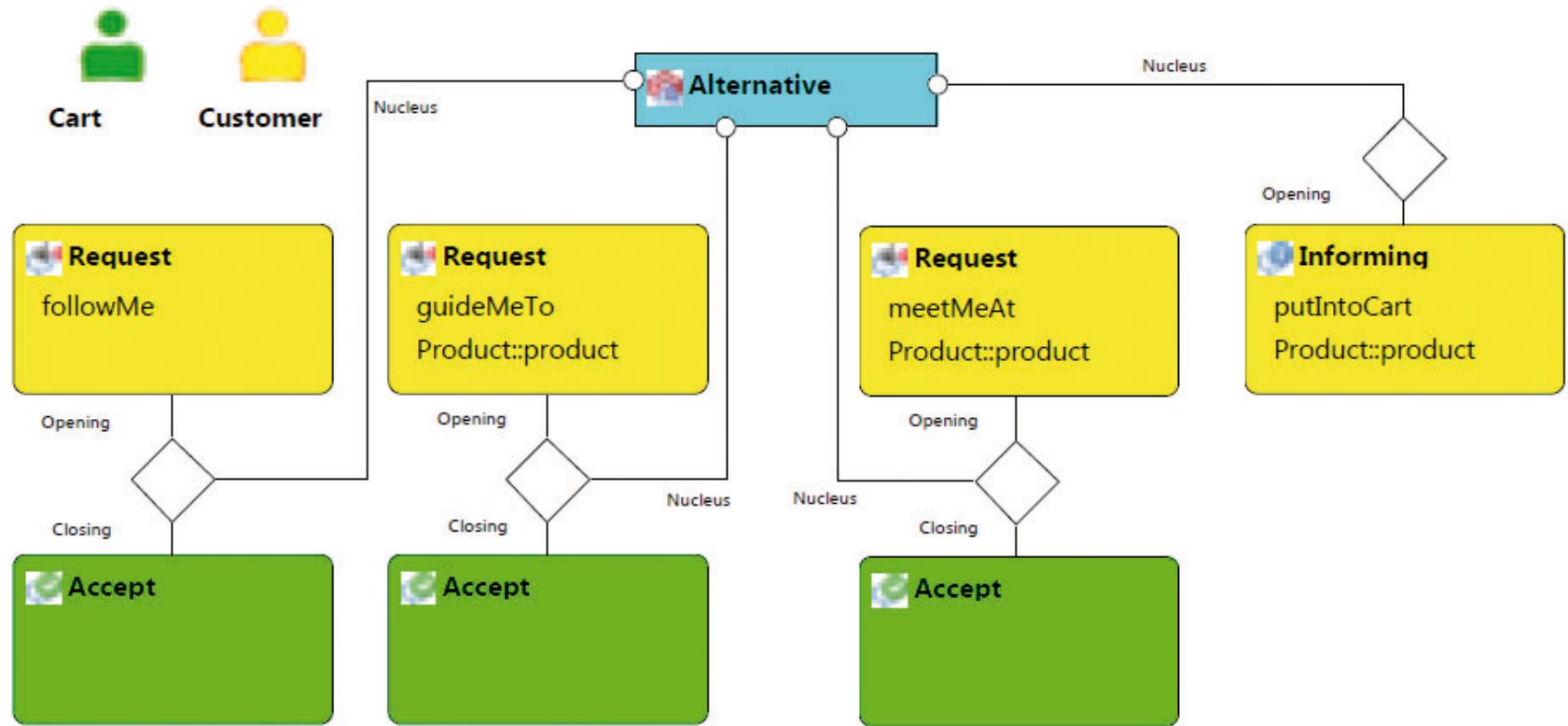
- Shopping Scenario as Running Example

Discourse-based Communication Models

- Discourse-based Communication Models as modality- and toolkit-independent basis for UI Generation
 - Domain-of-Discourse Model
 - Action-Notification Model – Natural Language Description for Commands
 - Discourse Model
- Communication Model specifies Propositional Content for each Communicative Act.
- Propositional Content refers to concepts from Domain-of-Discourse and Action-Notification Model
 - z.B.: *MeetMeAt param::destination*

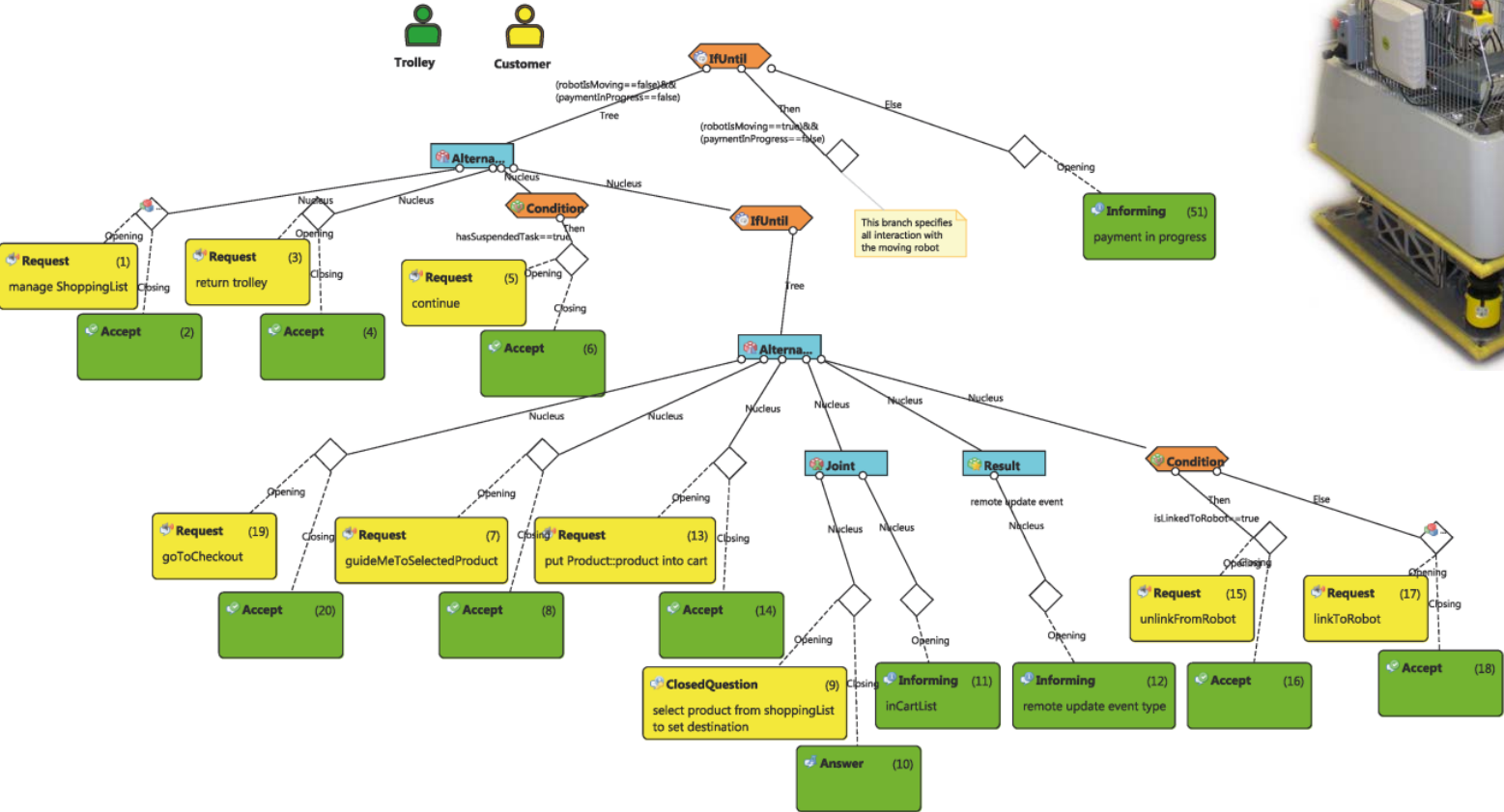
Communication Model Example

- CommActs are annotated with modalities
- meetMeAt Product::product



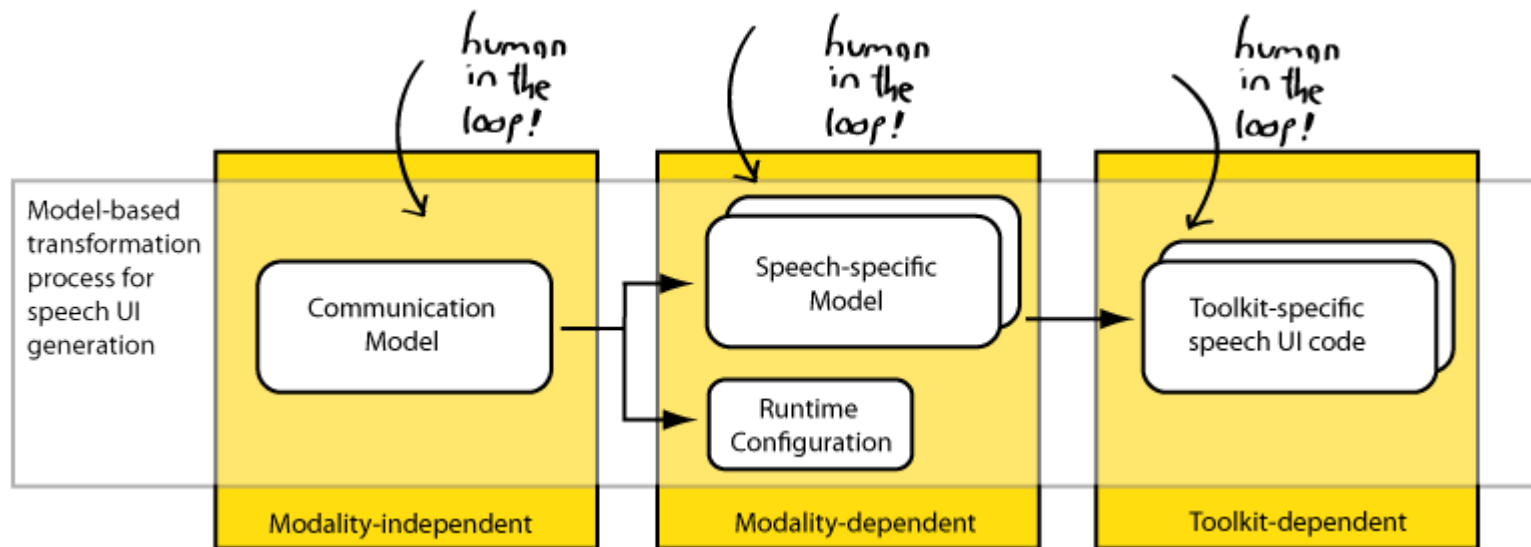
Shopping Example

- Shopping Robot

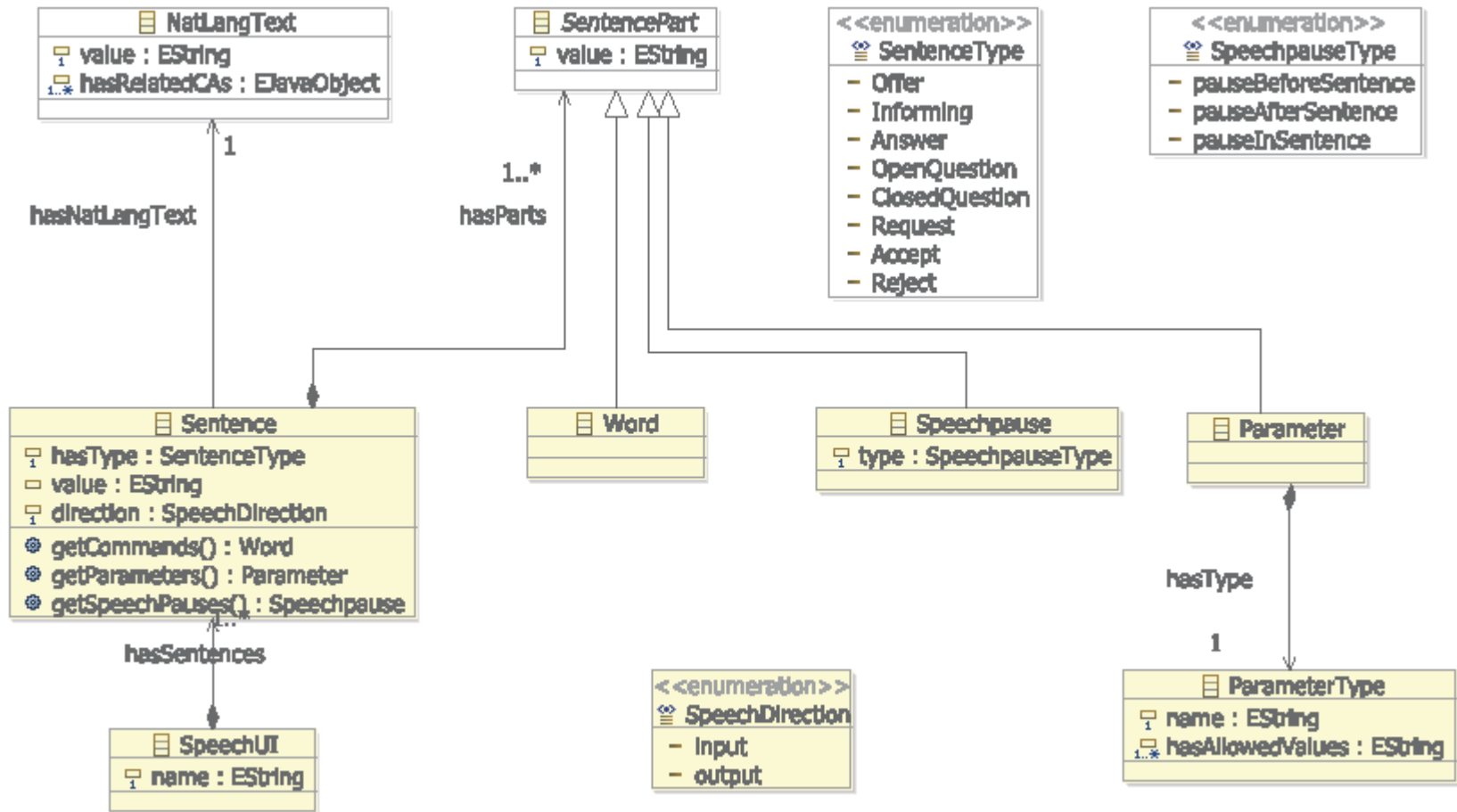


Semi-Automatic Generation Approach

- Full automation possible -> resulting speech UI not satisfying
- Keep the designer in the loop



Speech Metamodel



Speech Model Generation Algorithm

1. Extract modality dependent Communication Model
2. Use Action/Notification Description to create Natural Language Text for Sentence
3. Split Sentence according to delimiters into SentenceParts (i.e., Words, Speechpause or Parameter) and Sentence (i.e., Communicative) Act Type
4. Resolve Parameters according to Domain of Discourse Model elements (i.e., add all Products to the list of words)

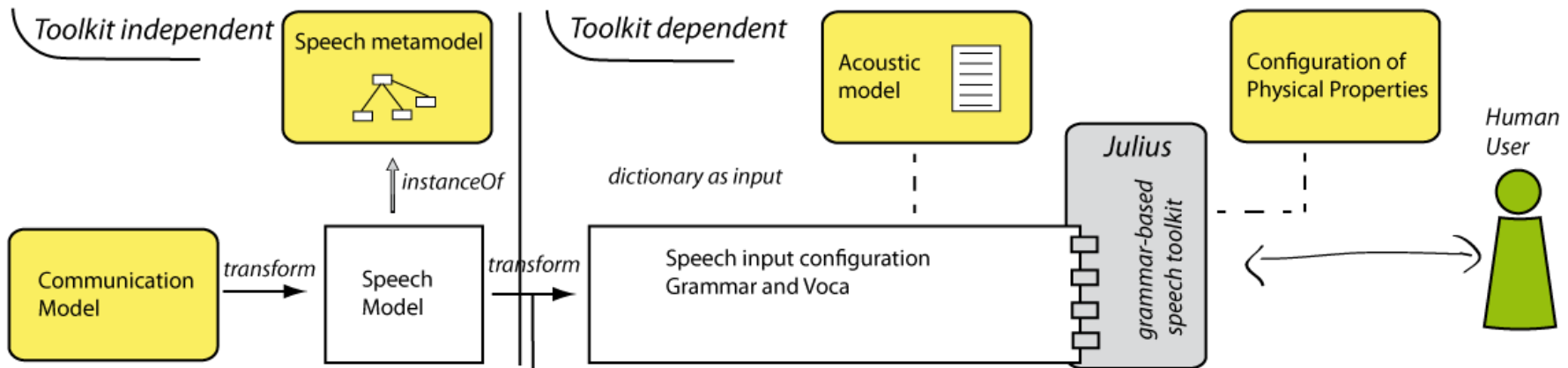
Generated Sentence

- Speech Input: Request Communicative Act from User
- Speech Output: Informing Communicative Act from Shopping Trolley
- Propositional Content: *MeetMeAt param::destination*
 - *MeetMeAt Speech Representation: Meet me at*

Type of SentencePart	Value of SentencePart
Speechpause	pauseBeforeSentence
Word	meet
Word	me
Word	at
Speechpause	pauseInSentence, value: 250 (if the sentence is rendered for speech output, the synthesizer waits 250 milliseconds before the following SentencePart is output)
Parameter	param::destination (the parameter can be of any value that is stored in the list of destinations in the Domain-of-Discourse Model)
Speechpause	pauseAfterSentence

Speech UI at Design Time

- Dictionary
- Acoustic Model
- Physical Properties for Julius Toolkit
- Concrete Speech Toolkits: Julius¹ (input), Festival² (output)



1: http://julius.sourceforge.jp/en_index.php
 2: <http://www.cstr.ed.ac.uk/projects/festival/>

Discussion



- Feasibility – simple, yet effective
- Support for different toolkits
- Command-based input and synthetization of output
- Natural Language description of commands must be provided
- Designer is enabled to compensate shortcomings of dictionary and acoustic model and check speech output realization



Conclusion



- Approach for semi-automatic speech UI generation based on Communication Models
- Exemplary Implementation for semi-autonomous shopping robot (for more information see <http://www.commrob.eu>)





Thank you for your attention!

