Introduction

- How to combine requirements in the form of use cases with interaction design
- How to automatically generate GUIs optimized for mobile devices from an interaction design
Combining Requirements and Interaction Design

Outline

- Background
- Interaction design based on discourse modeling
- Use case specification
- Sketch of automated user-interface generation
- Summary and Conclusion

Interaction design

- Design of interactions between human and computer
- Relation to requirements engineering
- Relation to task analysis
- No commitment to specific user interface
Communicative Acts – Open & Closed Question

- Open Questions enable asking for a particular type of information, respectively, an instance of a domain class.
- Closed Questions restrict the possible answer to a list of provided domain instances to choose from.
Institute of Computer Technology

Communicative Acts Taxonomy

- Assertive
- Commissive
- Directive
- Informing
- Offer
- Request
- Question
- Answer
- Opening
- Closing

Adjacency Pair

- Relates an initial communicative act with one subsequent communicative act or two alternative subsequent communicative acts.

- Typical adjacency pairs of communicative acts are:
  - ClosedQuestion–Answer, OpenQuestion–Answer
  - Offer–Accept, Offer–Reject
  - Request–Informing, Request–Accept, Request–Reject
Taxonomy of “Rhetorical” relations

Procedural construct – IfUntil

- If-statement combined with a conditional loop
- Utterance of the <Then> subtree depends on successful execution of the related Condition.
- Repetition of the <Tree> branch until Condition becomes fulfilled, while RepeatCondition is fulfilled
Procedural construct – IfUntil (cont.)

Flight Selection Discourse Model
Domain representation

- Speech act usually talks about something in the domain of discourse
- Part of model of the domain

Flight Booking Domain-of-Discourse Model

```
Flight
- number: EString
- date: EDate

Airport
- name: EString
- airportcode: EString

<enumeration>
- CreditCardProvider
  - MasterCard
  - VISA

CreditCard
- name: EString
- validThru: EString
- number: EInt
- cvc: EInt
- provider: CreditCardProvider

Ticket
- number: EString

Passenger
- name: EString
```
Combining Requirements and Interaction Design

**Use-case diagram**

- Bank
- Customer
- Withdraw Cash
- Deposit Cash
- Transfer between Accounts
- Identify Customer
- «include»

**OOA model – UML sequence diagram**

- Represents a scenario
- Interaction of instances
- Activation
- System border
Specification based on discourse model

- Scenario: focus on thread of events or actions
- Difficult to specify variations in Use-Case Report
- Discourse model: specification of class of dialogues
- Possible flows well defined and understandable
- Additional information in RST relations

Rendering of Final User Interfaces

- Automated generation of final (multimodal) UIs
- Generation of GUIs
  - Generation of Structural UI Model
  - Optimization (for Smartphones)
  - Generation of Behavioral UI Model
  - Weaving of Structural and Behavioral Models
- Even for multiple platforms
Weaving of Structural and Behavioral Models

- Different levels of abstraction

Tailoring for Specific Device (e.g., Smartphone)

- Objectives:
  - Maximum use of the available space
  - Minimum amount of navigation clicks, and
  - Minimum scrolling (except list widgets)

- Heuristic search for optimization (Branch & Bound)
Examples of Final User Interfaces – Desktop and Smartphones

- Simple flight-booking GUIs tailored with different strategies:
  - [http://ontoucp.ict.tuwien.ac.at/UI/FlightBooking](http://ontoucp.ict.tuwien.ac.at/UI/FlightBooking)
  - [http://ucp.ict.tuwien.ac.at/UI/FlightBookingScrolling](http://ucp.ict.tuwien.ac.at/UI/FlightBookingScrolling)
- Vacation planning:
  - [http://ucp.ict.tuwien.ac.at/UI/accomodationBooking](http://ucp.ict.tuwien.ac.at/UI/accomodationBooking)
- Potentially different GUIs tailored through optimization for different smartphones (screens)

Summary and Conclusion

- Interaction design can be based on discourse modeling.
- These models can be viewed as specifying classes of scenarios, i.e., use cases.
- These models can also be used for generating user interfaces.
- Requirements meet interaction design to make applications both more useful and usable.
Thank you for your attention!

Selected work of this tutorial presenter

Selected work of this tutorial presenter (cont.)