SIMtelligence Designer/J
A Visual Language to Specify SIM Toolkit Applications

Carsten Schmidt  Peter Pfahler  Uwe Kastens  Carsten Fischer
Interactive applications for mobile phones can be implemented in Java.

- **Deficiencies of Java in this domain:**
  - Restricted language subset
  - Java abstractions are not sufficient for certain requirements

- **Objectives of DSVL:**
  - Reduce development and testing effort
  - Be usable by non-programmers (network service provider)
Application Domain (2)

SIM–Toolkit applications have a typical structure

1. Handle menu selection and other events
2. Provide interactive procedures using SIM–Toolkit functions (display text, select item, get input, send SMS, ...)

Horoscope

send EMail

Enter EMail address

Enter message

Sending message...

Backward move!
Concepts of the Language (1)

There are two major views:

**Top-level view**
- **Menu**
  - Tools
    - Restaurant finder
    - Currency converter
  - Fun
    - Horoscope
    - Guess the number
- **Events**
  - SMS Envelope Event
- **Sub-Procedures**

**Procedure view**
- **Select search kind**
  - by Name
  - by Category
- **Select category**
  - Italian
  - Chinese
  - Greek
- **Select search**
  - Name
  - Location
  - Category
- **Alpha**
  - Sending request...
  - TelNo: 555-123
  - Data: REST_BY_NAME name
Concepts of the Language (2)
Summary

- DSVL for SIM−Toolkit Applications
  - Menu and event registration
  - Constructs for interactive procedures
    (SIM−Toolkit commands, backward moves)

- Achievements of the DSL
  - Hides some deficiencies of Java−subset (e.g. string handling)
  - Raises abstraction level (e.g. menus, backward move)

- Achievements of the visual structure editor
  - Easier to learn and use (e.g. dialogs for toolkit commands)