
Plug-and-Play Composition of Features and Feature Interactions with Statechart Diagrams

Christian Prehofer
DoCoMo Euro Labs
München

Feature-oriented Design with Statecharts

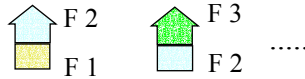
- We use UML statechart diagrams to model complex objects
- Modularity for statechart diagrams
 - Cross-cut complex statecharts into small parts (features)
 - Modular composition of statecharts from features
- Graphical description of features
 - Features & interactions are specified with partial statecharts
- Refinement rules for combining partial statechart diagrams
 - Based on refinement concepts for statechart diagrams
 - Interaction handlers (“adaptors”) as a glue for combining features
 - » Adaptors are represented graphically

Feature Combination (abstract view)

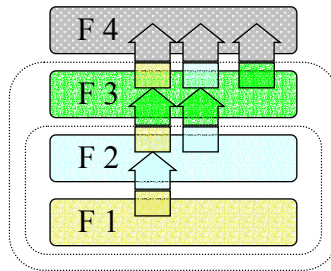
- Features (State + Methods)



- Adaptors



- Composition architecture



C. Prehofer

3

Example: Email Features

Features of an Email System (R. Hall, FIW 2000)

- Encryption of Emails
- Forwarding Emails
- Auto-reply
- Filtering of Emails
- Virus scanner
- ...

Feature Interactions

- Encryption and Forwarding
 - forward only encrypted
- Encryption and Auto-reply
 - Titel of email sent in plain as reply

C. Prehofer

4

Component-Design with Statecharts

- Statechart describes behavior of an object
 - Transition by external function call or internal action

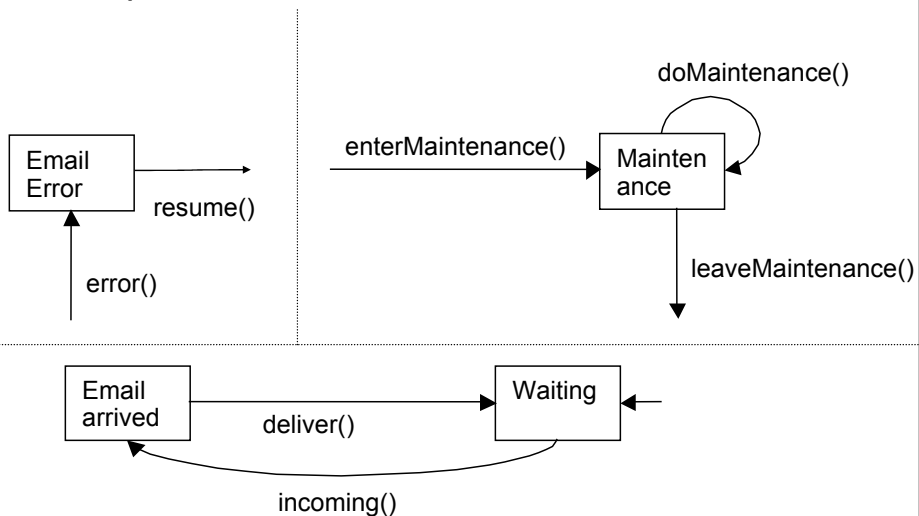
called_function() [condition] / action
→

- Composition of statecharts from Features

Object specification:	statechart
Feature:	partial statechart
Interaction handler (adaptor)	partial statechart
Feature composition:	statechart refinement
- Modular development of statecharts from features
 - Feature-interactions as statecharts-refinement

Feature model with partial statecharts

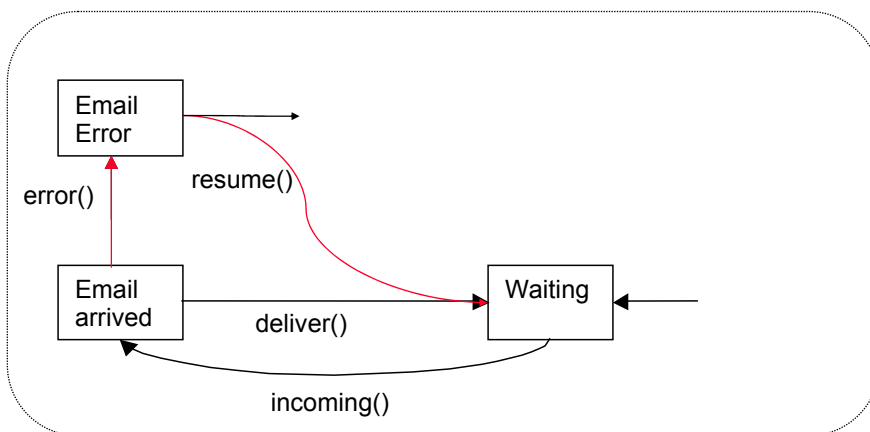
- Example: 3 features which add states



Classes of Features/Statecharts

1. Base features with a complete statechart
 - » includes an initial state.
 2. State-oriented features with a partial statechart
 - some states and transitions, but initial/final states not required
 - E.g. feature “MaintenanceMode” with one new state
 - » Only reachable by new transitions from other features.
 3. Transition-oriented features which define transitions
 - No (persistent) state
- Interaction specification („adaptors“) according to these classes
 - Combination rules according to these classes

Combination of 2 Features



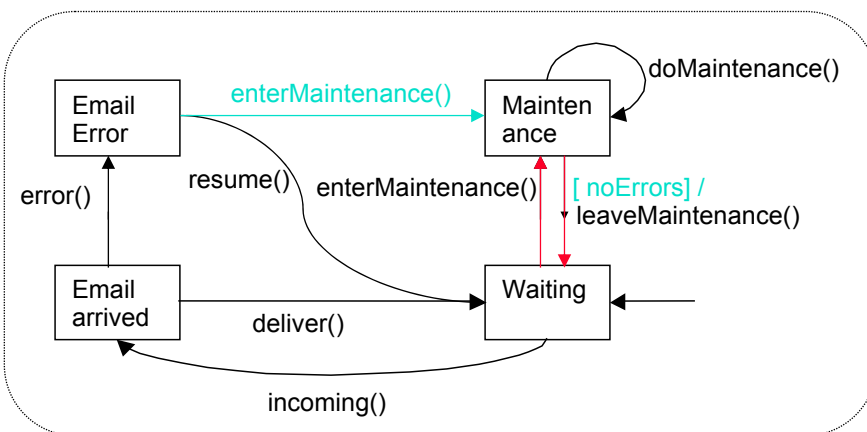
Composition as Semantic Refinement

- Use „Chaos“-Semantic based on external black box view
 - Permit as many traces as possible
 - If unspecified action occurs, anything is possible („chaos“)
- Refinement shall adds specific details
 - Reduces the number of possible traces
- Refinement steps for statechart diagrams
 - Add new behavior, which was not specified before
 - » e.g. add new states and „new“ transitions
 - Eliminate transitions if alternatives exits
 - Add internal or “compatible” behavior; do not change old behavior
 - » Need to abstract from new operations
 - » E.g. refine transition by statechart with internal transitions only
 - Eliminate/restrict transitions for exceptional cases
 - » Conditional refinement

C. Prehofer

9

Combination of 3 Features

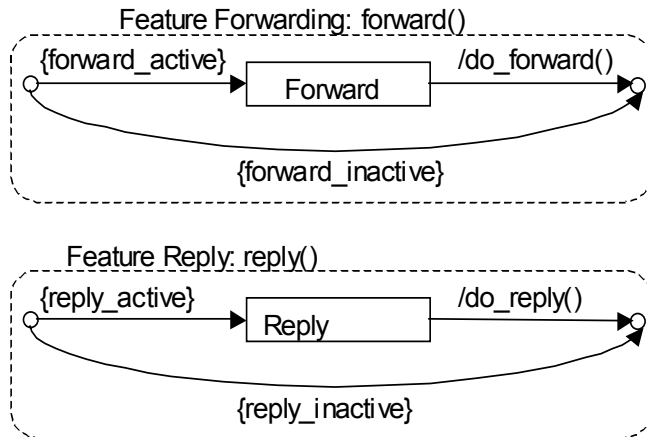


C. Prehofer

10

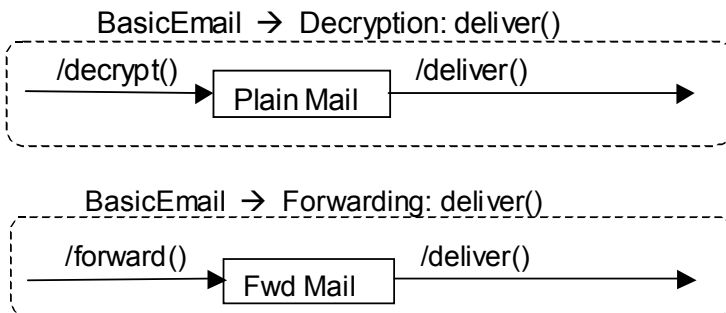
Feature Description (2)

- Forwarding and Auto-Reply Features
 - Transitions with internal actions



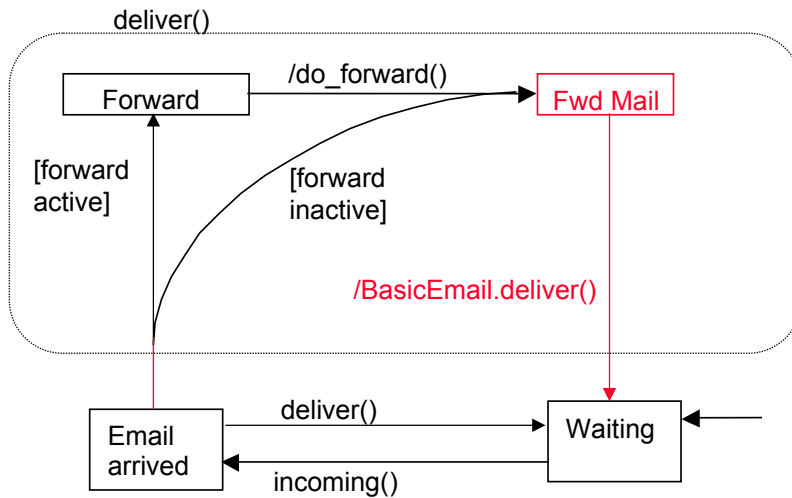
Combination Rules for Features

- Combination by refinement of transitions
 - Rules for adapting one feature to the other („interaction handling“)
BasicEmail → Decryption/Forwarding
 - Refinement describes the internal behavior of a transition by a statecharts



Transition Refinement

- Basic Email and Forwarding

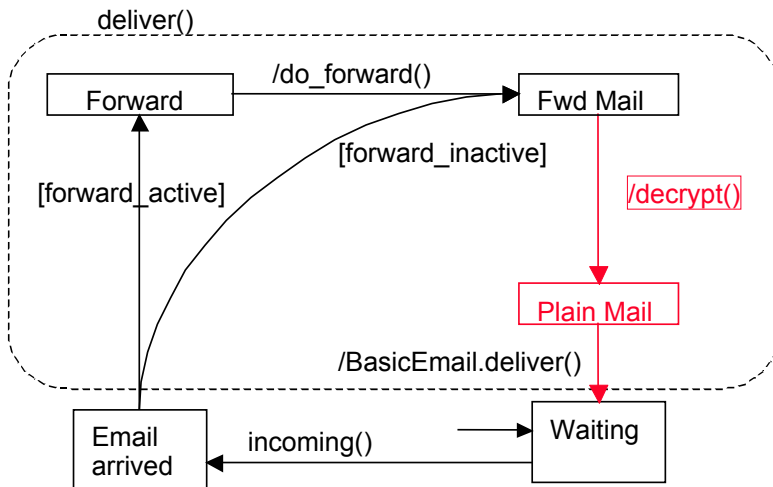


C. Prehofer

13

Combination of 3 Features (1)

- Basic Email, Forwarding and Decryption

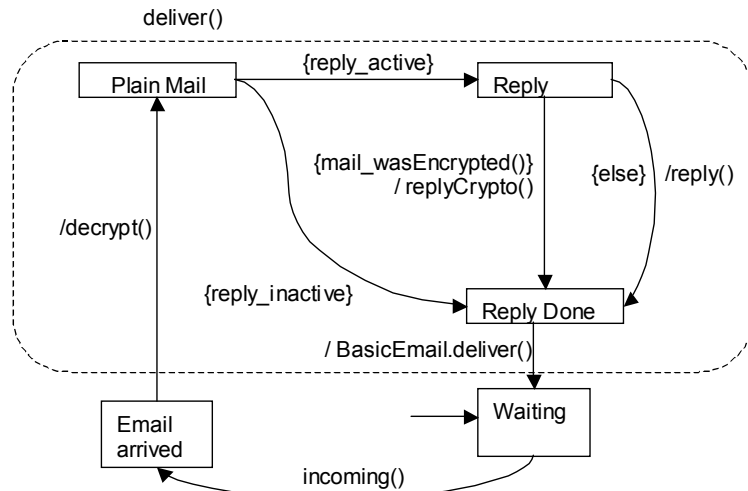


C. Prehofer

14

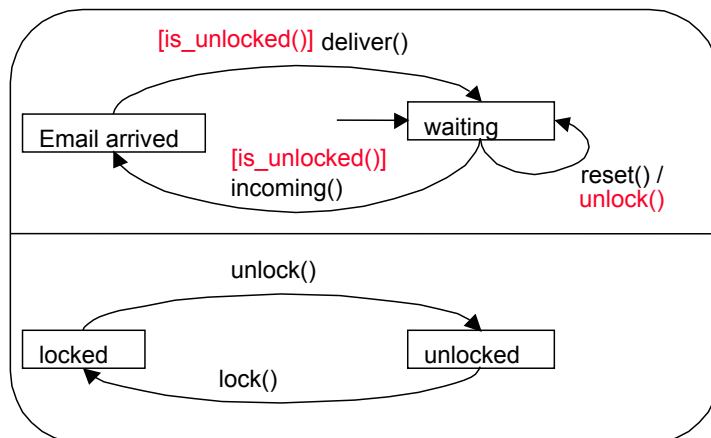
Combination of 3 Features (2)

- Basic Email, Auto Reply und Decryption



Combination Rules for Base Features

- Combination by parallel composition
 - Example: Lock-Feature with conditional refinement



Conclusions

- **Cross-cutting state description diagrams**
 - Modularity concepts for statechart diagrams
 - Supports flexible composition of features
 - Classify different kinds of partial statechart diagrams
- **Main benefit is graphical combination of features**
 - Compose just models of desired features
 - » Includes feature interactions
 - Different variations or extensions of a software component are needed.

Questions ?