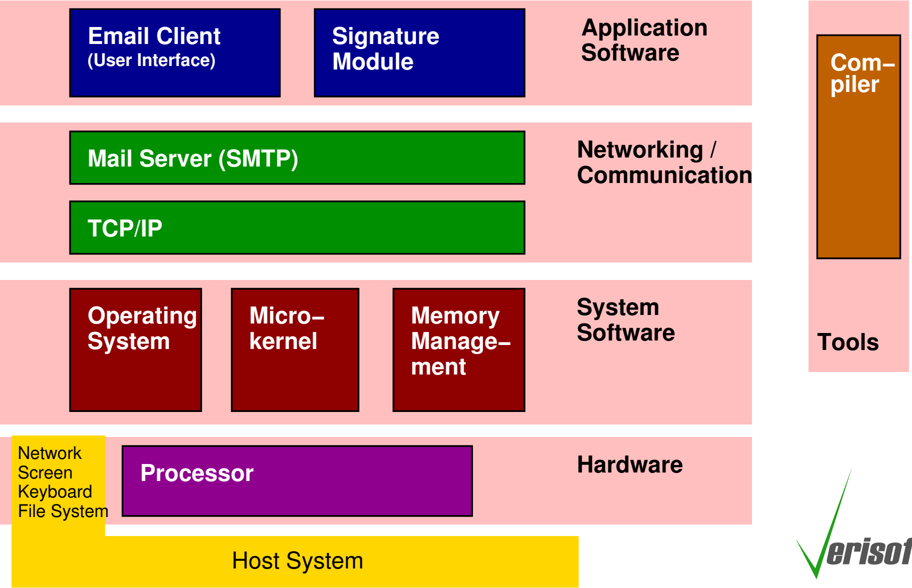
Multi-Formalism Specification and Verification in Verisoft

Thorsten Bormer

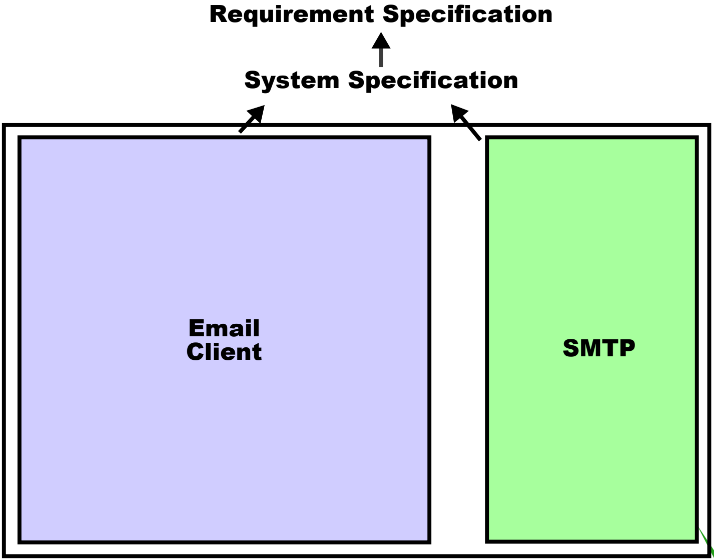
Universität Koblenz-Landau

June 15th, 2007

Component Overview



Specification/Implementation Layers



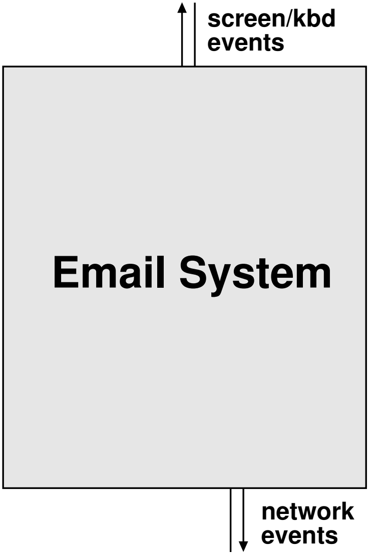
Requirement Specification

System Specification

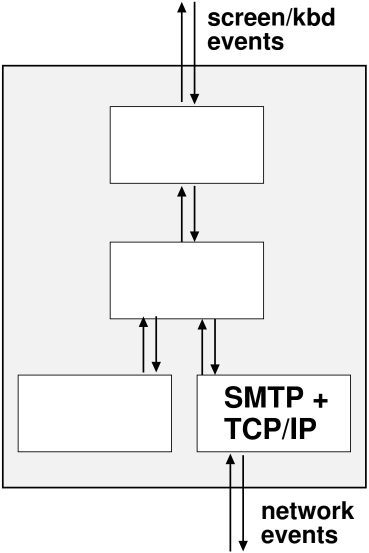
Email Client

SMTP

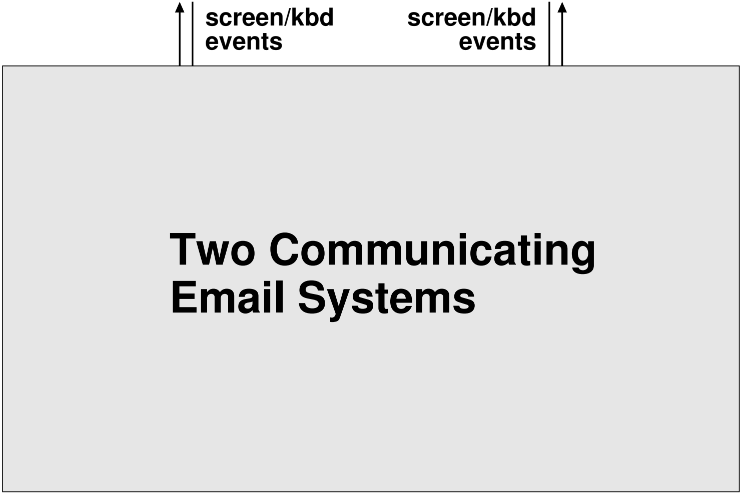
Top level Specification: One Email Client...



Top level Specification: One Email Client...

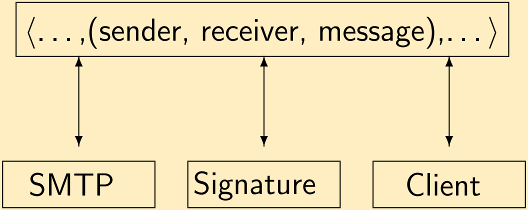


Top level Specification: ... or Two Clients?



Specification using Histories

Components communicate using events



- Specification on histories can be combined
- Computation of component is determined by events received

Example of Compontent specification

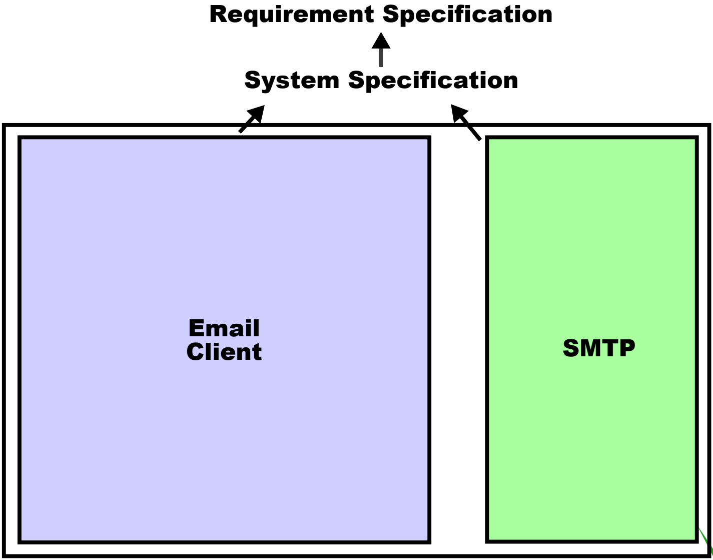
Example from the Component Specification of the Email Client:

"The User can enter any Email at will."

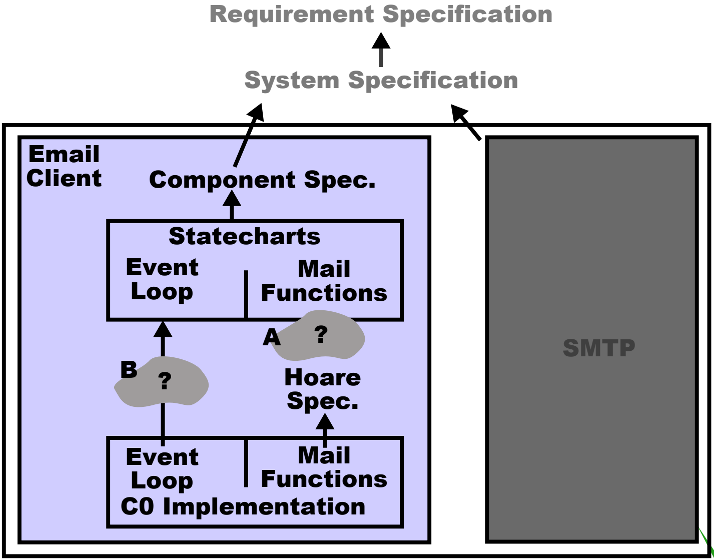
Let m be a string representing an email message.

$$\{h \mid h = h_{init} \circ h' \wedge \exists k.(h' \downarrow_{kbd,email} = k \wedge mailclientState(h').email = m)\}$$

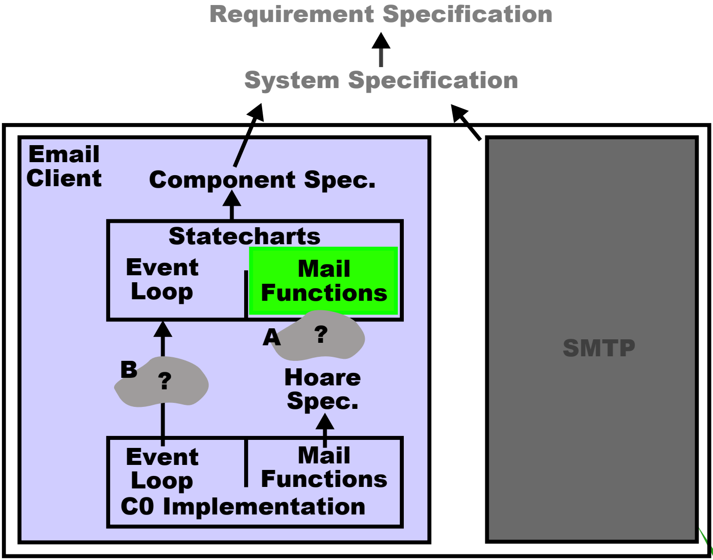
Specification/Implementation Layers



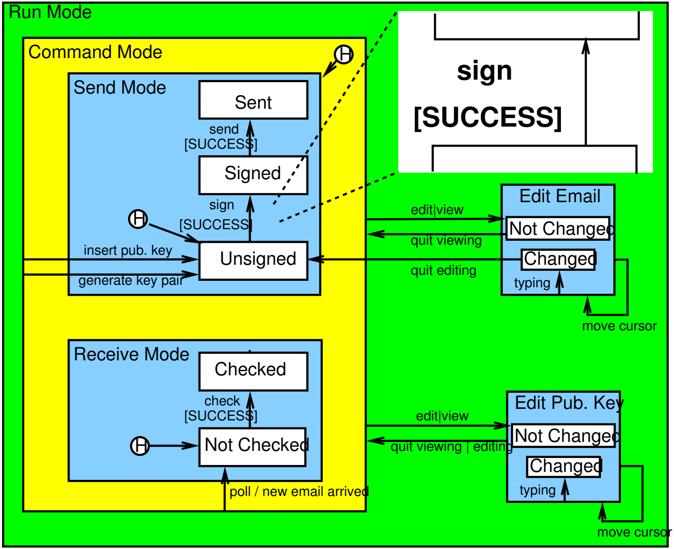
Specification/Implementation of the eMail Component



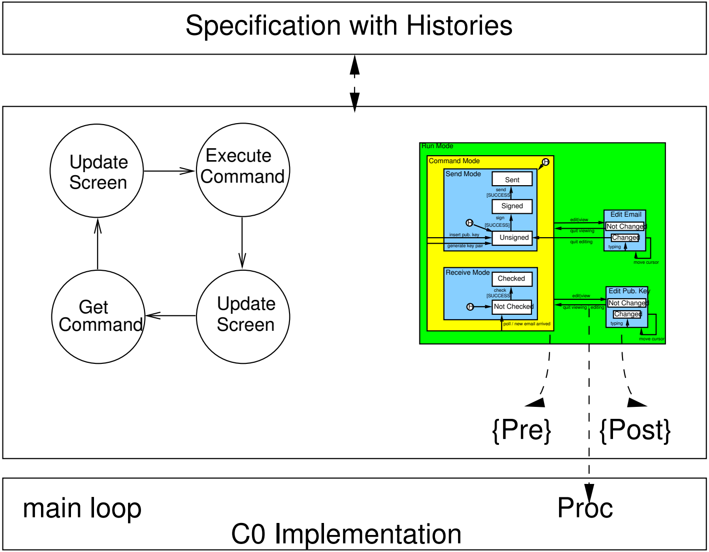
Specification/Implementation of the eMail Component



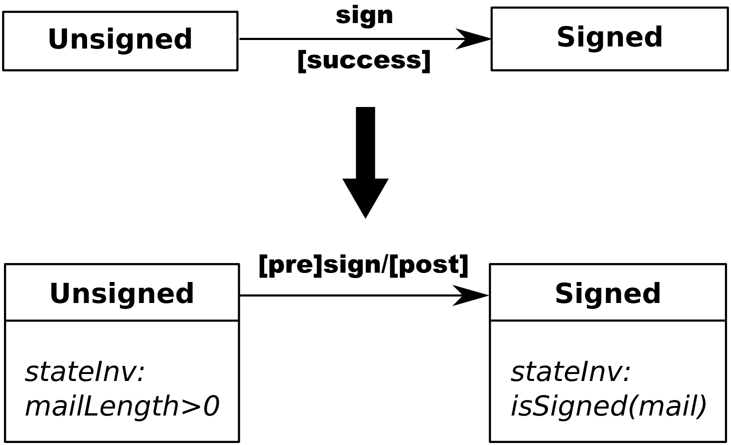
Specification with Automata



Relation: Implementation/Automata/Histories



Enhancing Automata with Structure



Unsigned

sign

[success]

Signed

Unsigned

[pre]sign/[post]

Signed

stateInv:
mailLength>0

stateInv:
isSigned(mail)

Implementation of the Event Loop

Event Loop

```
while(cmd != CMD_QUIT) {
        applicConfUpdateScreen(applicConf, osConf);
        osConfGetKeyStroke(osConf, key);
        cmd = command(*key,applicConf->state);
        applicConfUpdateScreen(applicConf, osConf);
        execute(applicConf,cmd,*key);
    }
```

Connection Automaton ↔ C0

Current Status

- verified that C0 implementation performs single transition in the statechart
- have to show that 'event loop' implements automaton

Verification of 'event loop'

- prove using Hoare-logic that one iteration always performs a valid transition
- prove using temporal logic that event loop implements automaton

Conclusion

- integration of specification/verification non-trivial task
- But: we're almost done!
- verification of the 'event loop' will be covered by my diploma thesis
- grateful for comments!

Thank you for your attention!