A Verification System for Distributed Objects with Asynchronous Method Calls

Wolfgang Ahrendt[1] Maximilian Dylla[12]

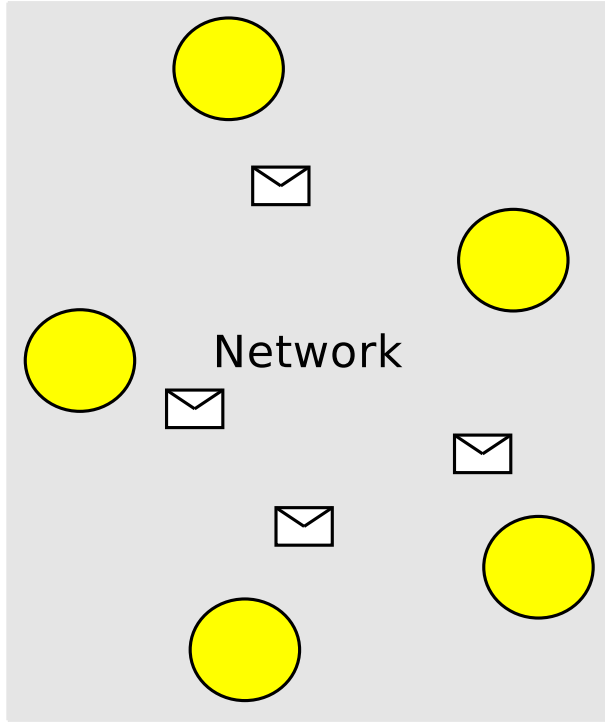
Chalmers University of Technology, Gothenburg, Sweden

Saarland University, Saarbrücken, Germany

Verification Subject: System Level

- language based on Creol
- distributed system of objects
- message passing
- communication via (co)interfaces
- asynchronous communication:

```
label ! obj.meth(x,y);
...;
label ? (z);
```

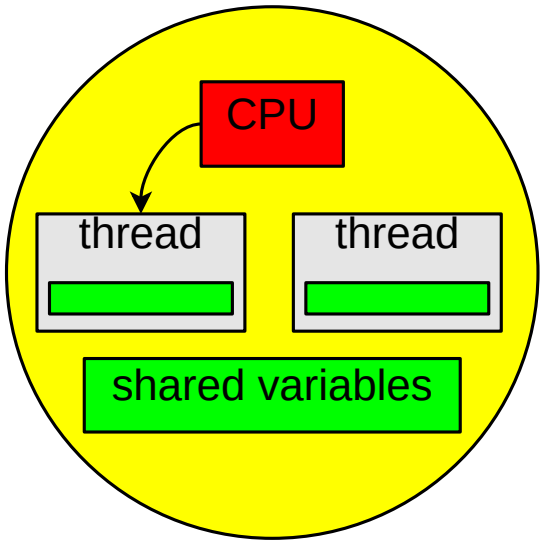


Verification Subject: Object Level

- one CPU per object
- method invocation: thread creation
- at most one active thread
- communication: shared variables
- cooperative scheduling
  $\Rightarrow$ release points:

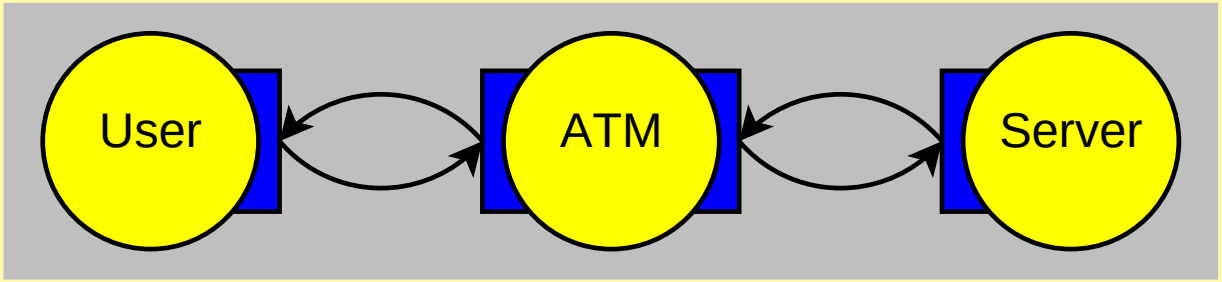
```
release
```

```
await exp
```



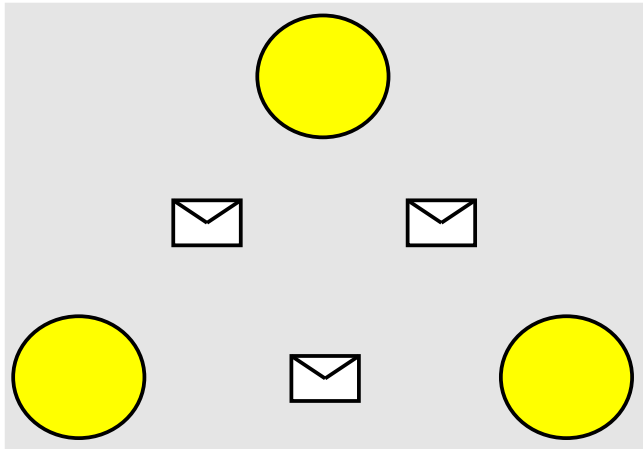
Example: Automated Teller Machine

```
interface S
 with ATM_S
  op authorize(in cardId:Int, code:Int;
               out ok:Bool)
  op debit(in cardId:Int, amount:Int;
           out ok:Bool)
end
```



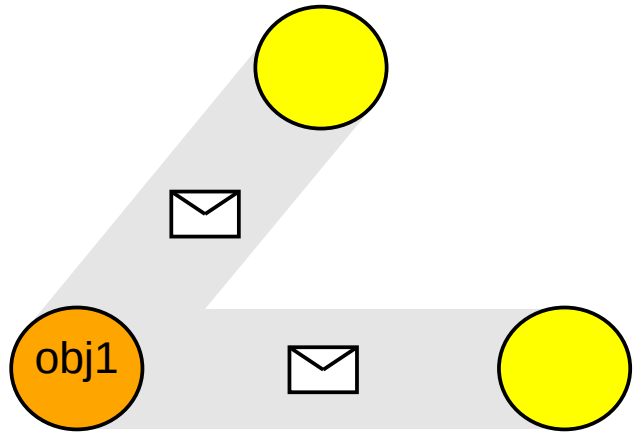
Specification: History

- system-wide ghost variable $\mathcal{H}$
- sequence of messages:
  - invocation
  - completion
  - object creation



Specification: History Projections

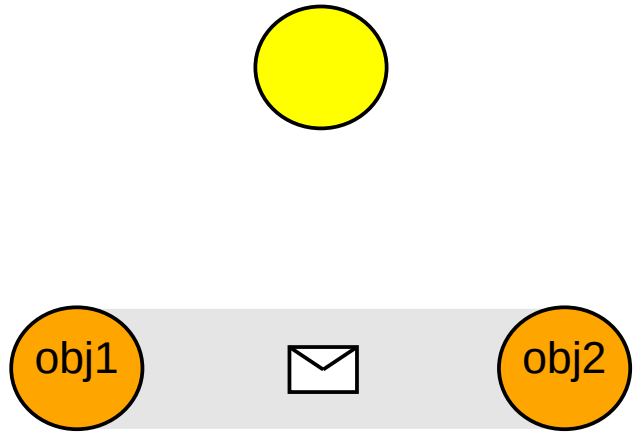
- Aim: modular verification
- messages sent or received by obj1: $\mathcal{H}/\text{obj1}$
- projection: $\mathcal{H}/\text{obj1} = \mathcal{H}/\text{obj1}/\text{obj1}$
- ensure well-formedness: $Wf(\mathcal{H}/\text{obj1})$



Specification: History Projections

- messages sent between obj1 and obj2:

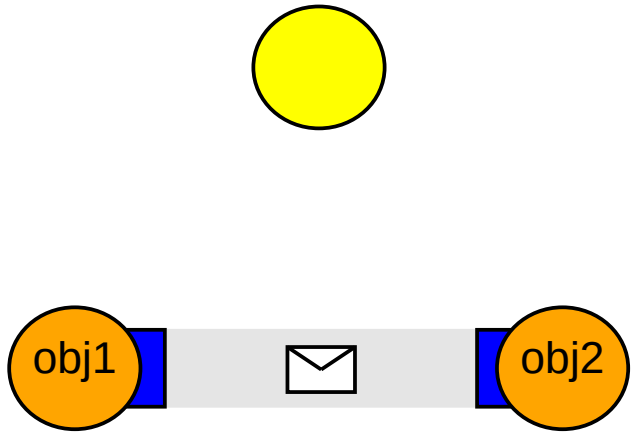
$$\mathcal{H}/\text{obj1}/\text{obj2}$$



Specification: History Projections

- messages sent between obj1 and obj2 via interface I:

$$\mathcal{H}/\text{obj1}/\text{obj2}/I$$



Example: Interface Specification

$$\mathcal{H}/\text{obj1}/\text{obj2}/\text{S} \leq \left(\text{auth}_{\rightarrow}(cid,.)\cdot\left(\text{auth}_{\leftarrow}(ff)\,\middle|\,\begin{pmatrix}\text{auth}_{\leftarrow}(tt)\\ \cdot\text{debit}_{\rightarrow}(cid,.)\\ \cdot\text{debit}_{\leftarrow}(.)\end{pmatrix}\right)\right)^{*}$$

- $\leq$: prefix
- $\rightarrow$: invocation
- $\leftarrow$: completion
- * : Kleene star
- $\cdot$: append
- (.) : wildcard for parameter

```
interface S
 with ATM_S
  op authorize(in cardId, . ;out ok)
  op debit(in cardId, . ;out ok)
end
```

Verification Process: Dynamic Logic

Dynamic Logic

- $\phi \rightarrow [p]\psi$
- forwards calculus:

$$\frac{\Gamma \vdash \{x := t\}[p]\psi, \Delta}{\Gamma \vdash [x := t; p]\psi, \Delta}$$

Hoare Logic

- $\{\phi\}p\{\psi\}$
- backwards calculus:

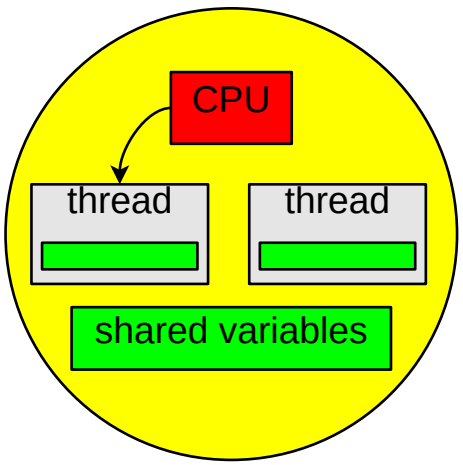
$$\frac{\{\phi[x/t]\}p\{\psi\}}{\{\phi\}p; x := t\{\psi\}}$$

- p : program statements
- ϕ : pre-condition
- ψ : post-condition
- Γ, Δ: prestate context
- $\{x := t\}$ explicit substitution

Verification Process: Object Level

$$\frac{\Gamma \vdash inv_c(\mathcal{H}, \overline{\mathcal{S}}), \Delta \qquad \Gamma \vdash \{U_{\mathcal{H},\overline{\mathcal{S}}}\}[p]\phi, \Delta}{\Gamma \vdash [\text{release; } p]\phi, \Delta}$$

- $\mathcal{H}$: history of the object $\mathcal{H}$/this
- $\overline{\mathcal{S}}$: shared variables
- p : following statements
- inv_c : class invariant
- $U_{\mathcal{H},\overline{\mathcal{S}}}$: overwrite history, shared variables
  $\mathcal{H}, \overline{\mathcal{S}} := \text{some } H, \overline{S}.(inv_c(H, \overline{S}) \wedge \mathcal{H} \leq H)$



Verification Process: System Level

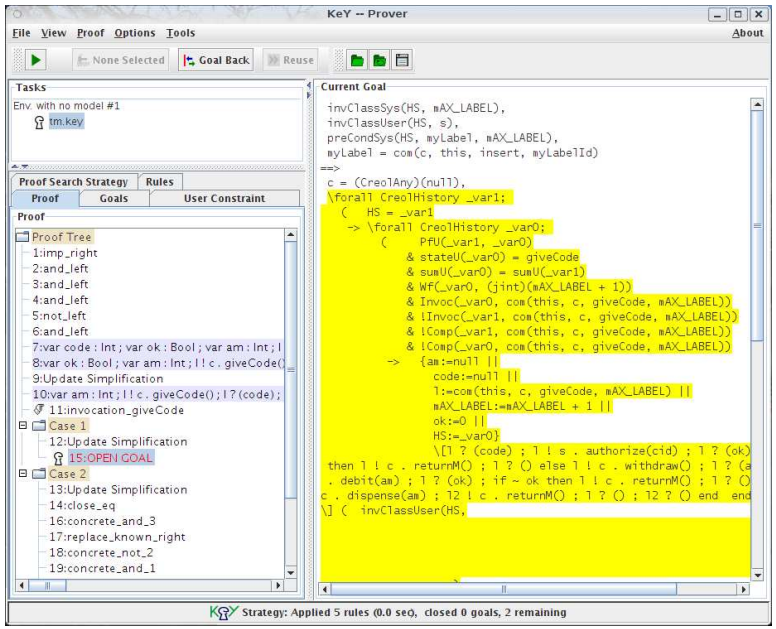
$$\frac{\begin{array}{l}\Gamma \vdash o = \mathsf{null} \rightarrow [\mathsf{block};\ p]\phi, \Delta \\ \Gamma \vdash o \neq \mathsf{null} \rightarrow \mathit{Wf}(\mathcal{H}) \wedge \mathit{inv}_I(\mathcal{H}/o/I), \Delta \\ \Gamma \vdash o \neq \mathsf{null} \rightarrow \{U_{\mathsf{label}}\}\{U_{\mathcal{H}}\}[p]\phi, \Delta\end{array}}{\Gamma \vdash [\mathsf{label}\ !\ o.\mathsf{mtd}(\bar{\mathsf{a}});\ p]\phi, \Delta}$$

- $\mathcal{H}$: history of the object $\mathcal{H}/\mathrm{this}$
- p : following statements
- inv_I : interface invariant
- U_{label} : reference to method call
  $\mathsf{label} := (\mathrm{this}, o, \mathsf{mtd}, \bar{\mathsf{a}}, i)$
- $U_{\mathcal{H}}$: append invocation message

$$\mathcal{H} := \text{some } H. \begin{pmatrix} \mathit{Wf}(H) \wedge \mathcal{H} \leq H \wedge \mathit{inv}_I(H/o/I, \bar{\mathsf{a}}) \\ \wedge \mathit{Invoc}(H, \mathsf{label}) \wedge \neg \mathit{Invoc}(\mathcal{H}, \mathsf{label}) \end{pmatrix}$$

The Prototype

- based on KeY
- interactive, automated theorem prover
- rules, problem files formulated as taclets
- creation of example specific rules necessary



Example: Proof Statistics

- 2495 proof steps (in total)
- 27 branches
- 10% of all steps interactive

Questions?