Workshop „Modellierung in der Hochschullehre“ (MoHoL)
im Rahmen der Modellierung 2022

eingeladener Vortrag

Modellierung in der Informatik – Hochschullehre

- ein Trauerspiel -

Wolfgang Reisig

Humboldt-Universität
zu Berlin
Germany

1. Die Lage

2. Ein Beispiel

HPI: Bubblesort https://hpi.de/friedrich/teaching/units/einfache-sortierverfahren.html

1. Acht Spielkarten werden sortiert

2. Pseudocode

```
function BubbleSort(A) {
  newRound := TRUE;
  for i := n-1 downto 1 do {
   if newRound then {
     newRound := FALSE;
     for j := 0 to i-1 do {
       if A[j] > A[j+1] then {
         temp := A[j];
         A[j] := A[j+1];
         A[j+1] := temp;
         newRound := TRUE;
       }
     }
   }
   else break
  }
}
```

„Bubblesort ist die Verschachtelung von zwei For-Schleifen“

„Die äußere For-schleife macht …

„Elemente sind falsch rum“

„Vergleich stimmt“

Vernünftig über bubblesort reden

Gegeben: Eine Menge A, eine Ordnung auf A

daraus abgeleitet: Tupel, geordn. Tupel, Liste, geordn. Liste

für Tupel (a, b):

„Tupel (a, b) ordnen"
ord: Tupel -> Tupel
ord(a, b) $=_{def}$ (a, b), falls (a, b) geordnet, sonst (b, a).

für Listen (a0 … an):

„(ai-1, ai) ordnen":
i-Schritt: Listen -> Listen
i-Schritt(a0 … ai-1, ai, … an) $=_{def}$ (a0 … ord(ai-1, ai) … an)

Wir schreiben „i-Schritt" statt „i-Schritt(a0 … an)",
wenn (a0 … an) im Kontext klar ist.

„Das größte Element ganz nach rechts spülen":
max: Listen -> Listen
max(a0 … an) $=_{def}$ 1-Schritt; 2-Schritt; …; n-Schritt.

„ai *an die richtigen Stelle setzen, wenn* ai+1 … an *schon an der richtigen Stelle stehen":*
i-ordnen: Listen -> Listen
i-ordnen(a0 … ai-1, ai… an) $=_{def}$ (max(a0 …ai) ai+1 … an)

Wir schreiben „i-ordnen" statt „i-ordnen(a0 … an)",
wenn (a0 … an) im Kontext klar ist.

„(a0 …, an) ordnen":
geordnet: Listen -> Listen
geordnet(a0 …, an) $=_{def}$ n-ordnen; n-1-ordnen; …; 1-ordnen.

Was lehrt uns das?

An den vielen Studienabbrechern sind wir Schuld!

[Pflüger 94] zur Informatik-Ausbildung:

Fertigkeiten ersetzen Wissen,

Machen erdrückt das Verstehen.

Typischerweise lernen die meisten, Programme zu schreiben,

ohne jemals welche gelesen, geschweige verstanden zu haben.

Historische Entwicklung

[Floyd 67]: *Assigning Meanings to Programs.*

[Knuth 68]: *The Art of Computer Programming*

synthetische Programmiersprache

Heute

Was bietet die Modellierungs-Community?

Vorschläge der Softwaretechnik: *Abstract State Machines (ASMs) / Actor model / Alloy / ANSI/ISO C Specification Language (ACSL) / Autonomic System Specification Language (ASSL) / B-Method / CADP / Common Algebraic Specification Language (CASL) / Esterel / FOCUS / Java Modeling Language (JML) / Knowledge Based Software Assistant (KBSA) / Lustre / mCRL2 / MSC-LSC / Perfect Developer / Petri nets / Predicative programming / Process calculi: CSP, LOTOS, π-calculus / RAISE / Rebeca Modeling Language / SPARK Ada / Specification and Description Language (SDL) / Statecharts / TLA+ / USL / VDM: VDM-SL, VDM++ / Z notation*

Vorschläge der Wirtschaftsinformatik: *ADONIS / ARIS / BPMN / EPK / MEMO / St. Gallen Approach / UML-Varianten*

Systematik dahinter?

Anforderungen an eine Basis der Modellierung

Schluss
Thesen zur Modellierung in der Informatik-Lehre

[Pflüger 94]:

Es kann keine Rede davon sein, dass sich aus der Technik des Formalen die Grenzen der informatischen Modellierung verstehen lassen.

kurz und gut

aus Sicht der Anwendung modellieren!
daraus code generieren

nicht: aus Sicht der Implementierung programmieren

Literatur

[Dijkstra 89] Dijkstra, E. W. *Reply to comments*. Commun. ACM, 32(12) (1989). 141 - 144.

[Floyd 67] Robert W. Floyd: *Assigning Meanings to Programs*. Proceedings of the Symposium on Applied Mathematics Vol 19 American Mathematical society 1967, pp 19 - 32

[HPI 22] https://hpi.de/friedrich/teaching/units/einfache-sortierverfahren.html

[Knuth 68] Donald E. Knuth: *The Art of computer Programming*. Addison-Wesley 1968

[Pflüger 94] J.Pflüger: *Informatik auf der Mauer* in: Informatik- Spektrum (1994) 17:251-257

[Reisig 20] Wolfgang Reisig: *Informatics as a Science*. Enterp. Model. Inf. Syst. Archit. Int. J. Concept. Model. 15: 6:1-6:13 (2020)

? ? ∈ ? × ? ? ?? ?
??? ??????? ?
?????????
∀∃⇐*√⊗

aa

?

¬ ∧ ∨ ? ??

→ ↔ ? —

± ? * ∞ ≥ ≤ < > ≠ ≡ ~ ≈ ≅

⊂⊆ ∈ ∉ ⊃ ⊇ ⊄ ∩ ∪ ⊆ × ? ? ??

⊕ ⊗ M^{ω} M^{*} M^{∞}

λ π τ μ ω σ ρ Δ Ξ Π Λ Σ Φ Ω

? ? ^ ??????

??????? ?

?????

HERAKLIT **A^{*} A^{+} A^{-} B^{*} B^{+} B^{-}**

$A^{*}A$ $A^{+}A$ $A^{-}A$ $B^{*}B$ $B^{+}B$ $B^{-}B$