

# Sichere Software

Werner Stephan

Source: Stephan



# Software-Krise 1969

## NATO Konferenz: Software-Engineering

- Phasen-Modelle
- Projektmanagement
- Qualitätssicherung
- Requirements-Engineering
- Case-Tools
- UML



# Softwarekrise 1999

Bedarf<sup>(1)</sup> an sicherer<sup>(2)</sup> Software:

- Eingebettete System: Automobile, Eisenbahn, ...
- IT-Sicherheit: Chipkarten, Digitale Signaturen, Transaktionen über offene Netze

(1) neu: kommerzieller Hintergrund

(2) betrifft Inhalt: „Software, die tut was sie soll.“



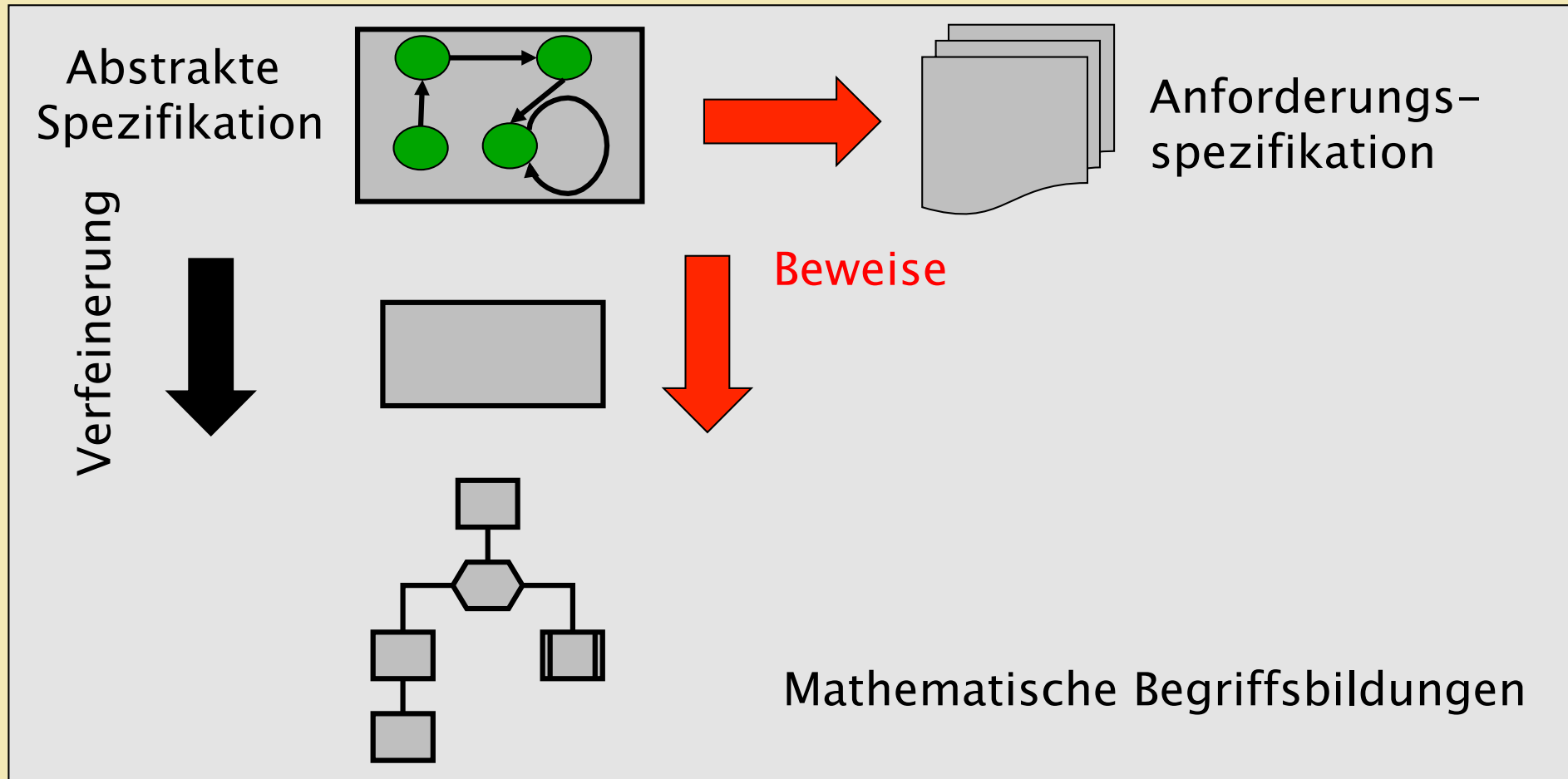
# Softwarekrise 1999

## Lösung: Verwendung formaler Methoden

- Beschreibung des Ziels: Was soll erreicht werden?
- Beschreibung des Wegs: Wie kommt man zu einer korrekten Lösung?



# Formale Methoden



Source: Stephan



# Formale Methoden

Math. Modellbildung: präzise Abstraktionen



— Beschreibung in

Logik: Schlußfolgerung



— Realisierung durch

Systeme: Durchführung

# Beispiel: Notschliesssystem (NSS)



Source: Stephan

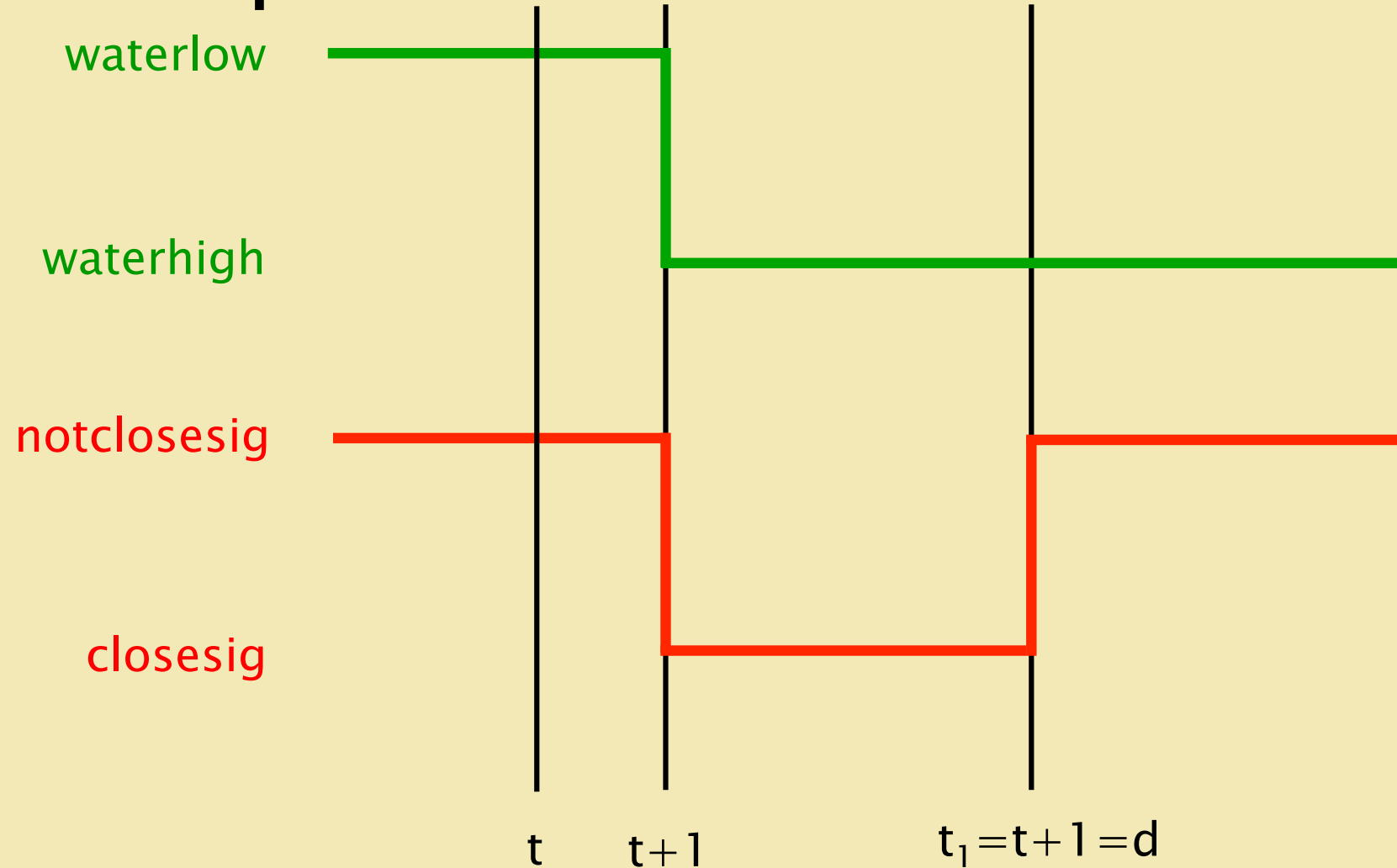


# Notschliesssystem (NSS)

- Storm Surge Barrier, Osterschelde, Holland
- Tore müssen bei kritischem Wasserstand schliessen
- Messstationen zur Erfassung der Wasserstände
- Kontrollsystem zum Steuern der Tore
- Realzeit: Komplexes Zeitverhalten



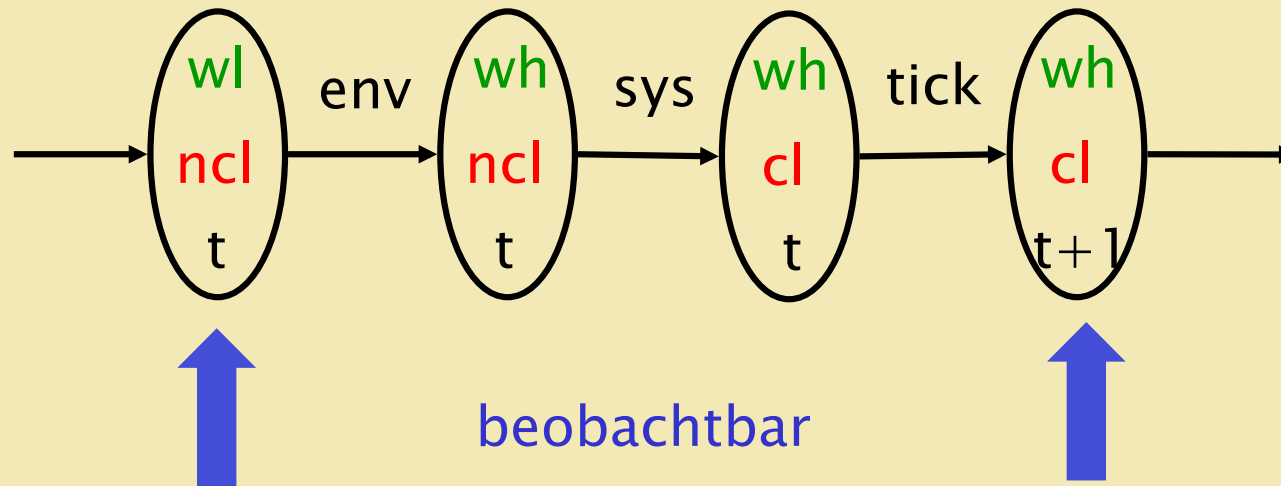
# Beispiel für einen zeitlichen Ablauf



Source: Stephan

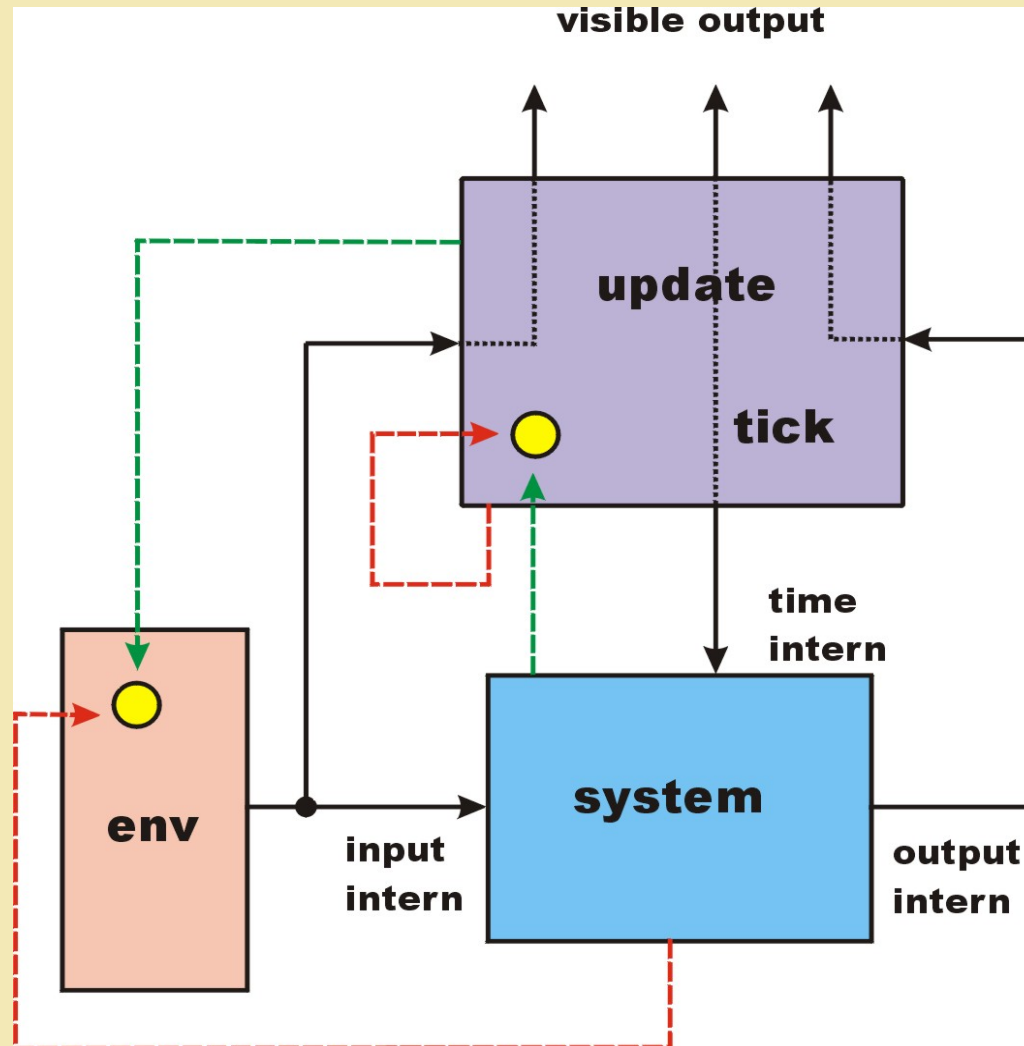


# Berechnungsfolgen (Traces)



Verschränkte Ausführung von System – Umgebung – Uhr

# Struktur des NSS-Modells

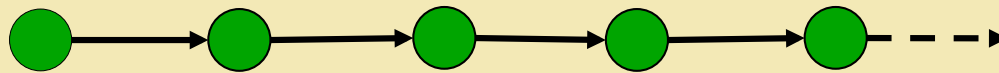


Source: Stephan

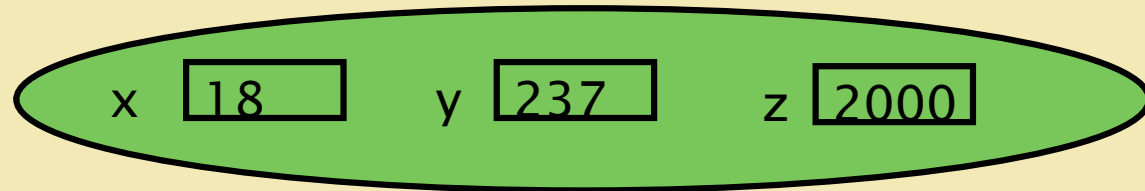


# Temporallogische Formalisierung

- Beschreibung von Ausführungssequenzen:



- Zustände



- Operatoren:  $\square$  always,  $\diamond$  eventually, **unless**
- Schritte (Aktionen):  $x' = x + 1$
- Systeme:  $\square (A_1 \vee \dots \vee A_n) \wedge \text{INIT} \wedge \text{FAIR}$

# Formale Methoden im Safety-Bereich

## 1. Notschliesssystem Osterschelde (NSS)

- Beschreibung
- Zeit- und Strukturmodell
- Eigenschaften und Ergebnisse

## 2. Robertino Control System

- Beschreibung
- Sicherheitskritische Komponenten
- Sicherheitsmodell und Ergebnisse



# Allgemeines Zeitmodell

- Globale Zeit als natürliche Zahlen realisiert  
⇒ diskrete Zeitskala
- Sichtbare Variablen werden nur angeschaut, wenn sich die Zeit ändert
- Signallängen werden durch “rigide” Variablen beschrieben
- Granularität im Nachhinein festlegbar  
⇒ Systembeschreibung für bel. Signallängen



# Ergebnisse

- Schreibfehler in der Beschreibung mit semantischen Auswirkungen
- Unterspezifizierte und überflüssige Transitionen in der Orginalspezifikation
- **Deadlock in der Zustandsmaschine**



# Robertino

- Am JRC (Joint Research Center), Ispra, entwickelt
- Austausch von Kernummantelungssegmenten in einem Fusionsreaktor
- Projekt zur Design-Validierung (ferngesteuerter) Roboter
- Positionierung der Segmente mit einer Genauigkeit von 0,1 mm und 0,01 deg
- Maximale Geschwindigkeit:  
x-, y-, z-Achsen: 20 mm/s  
w-Achse: 2 deg/s



# Robertino Control Software

- RCS ist der Kern des Kontrollsystems zur Steuerung der Achsenbewegungen
- RCS implementiert in C (50 kLOC) unter VxWorks
- Round Robin Scheduling (Zeitscheiben)
- Tasks sind preemptiv (Prioritäten)
- RCS besteht aus mehreren Komponenten  
→ **FLD-Interface**, **Interpolator**, CPI



# Sicherheitseigenschaften

## Informal:

- Robertino bewegt sich in die richtige Richtung bei Anliegen eines move-Kommandos
- Workspace limits: Robertino bleibt immer innerhalb seiner Arbeitsraumbegrenzung
- Beschleunigung: Robertino überschreitet nicht die vorgegebenen Beschleunigungs- und Bremslimits



# Ergebnisse

- Formale Spezifikation für nebenläufige Prozesse
- 2000 Zeilen Spezifikation in VSE II
- Realzeit explizit spezifiziert
- Formaler Entwicklungsprozess:
  - Unvollständige Transitionstabellen
  - Fehlerhafte Verhaltensbeschreibungen (Grenzwertfehler)

