Sichere Software

Werner Stephan

Source: Stephan

Software-Krise 1969

NATO Konferenz: Software-Engineering

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

Source: Stephan

Softwarekrise 1999

Bedarf[(1)] an sicherer[(2)] 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."

Source: Stephan

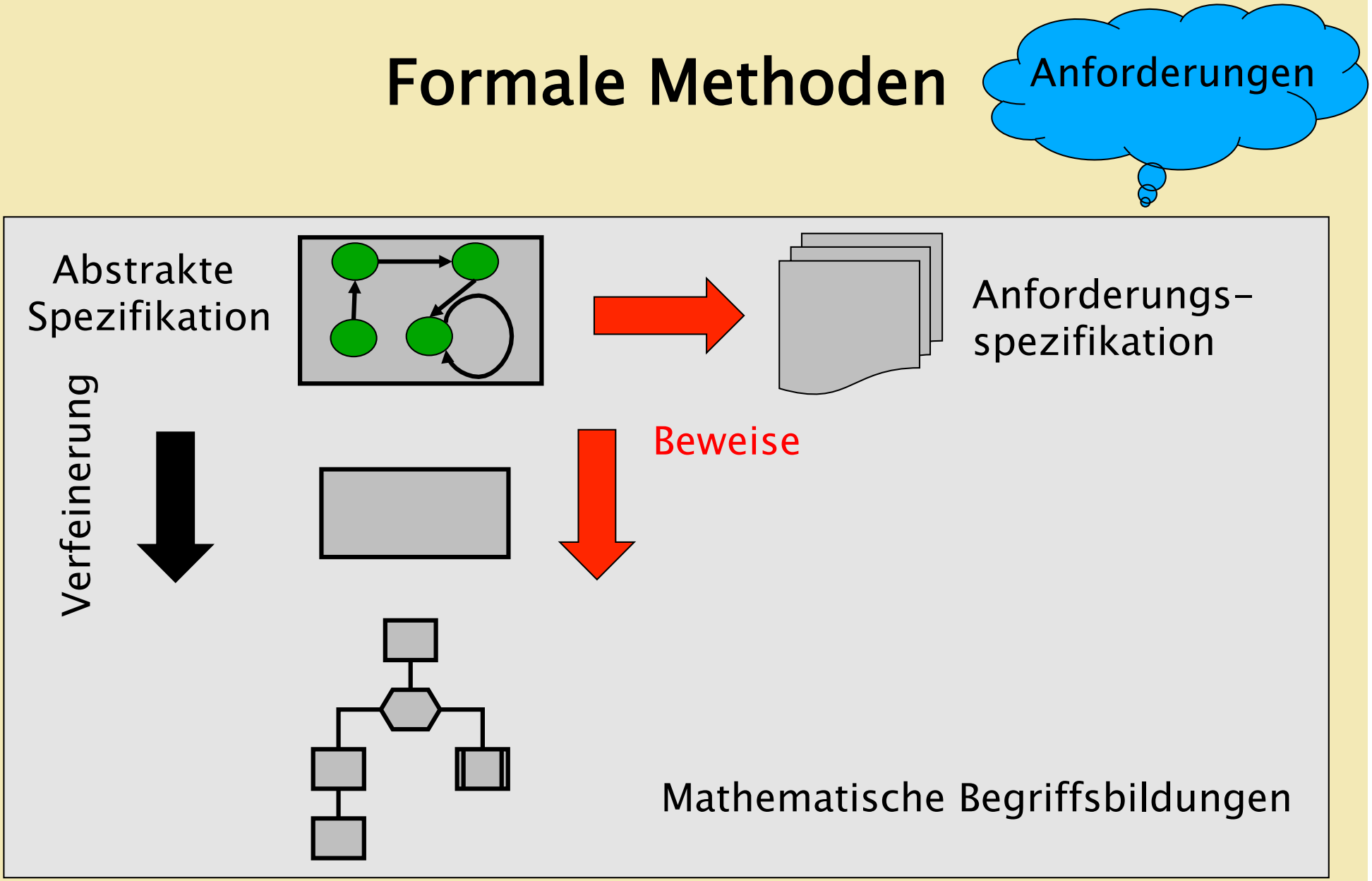
Softwarekrise 1999

Lösung: Verwendung formaler Methoden

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

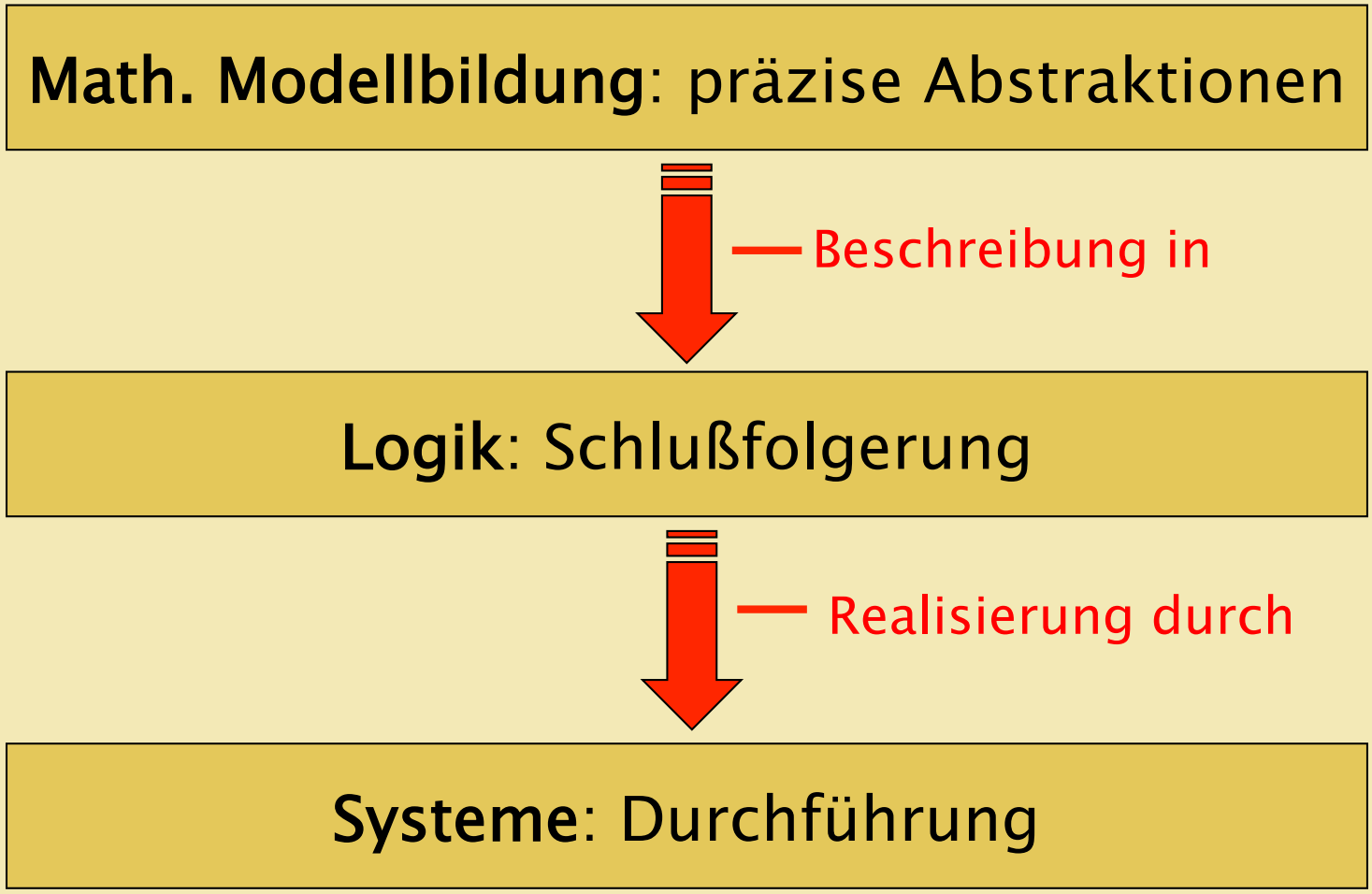
Source: Stephan

Formale Methoden



Source: Stephan

Formale Methoden



Source: Stephan

Beispiel: Notschliesssystem (NSS)

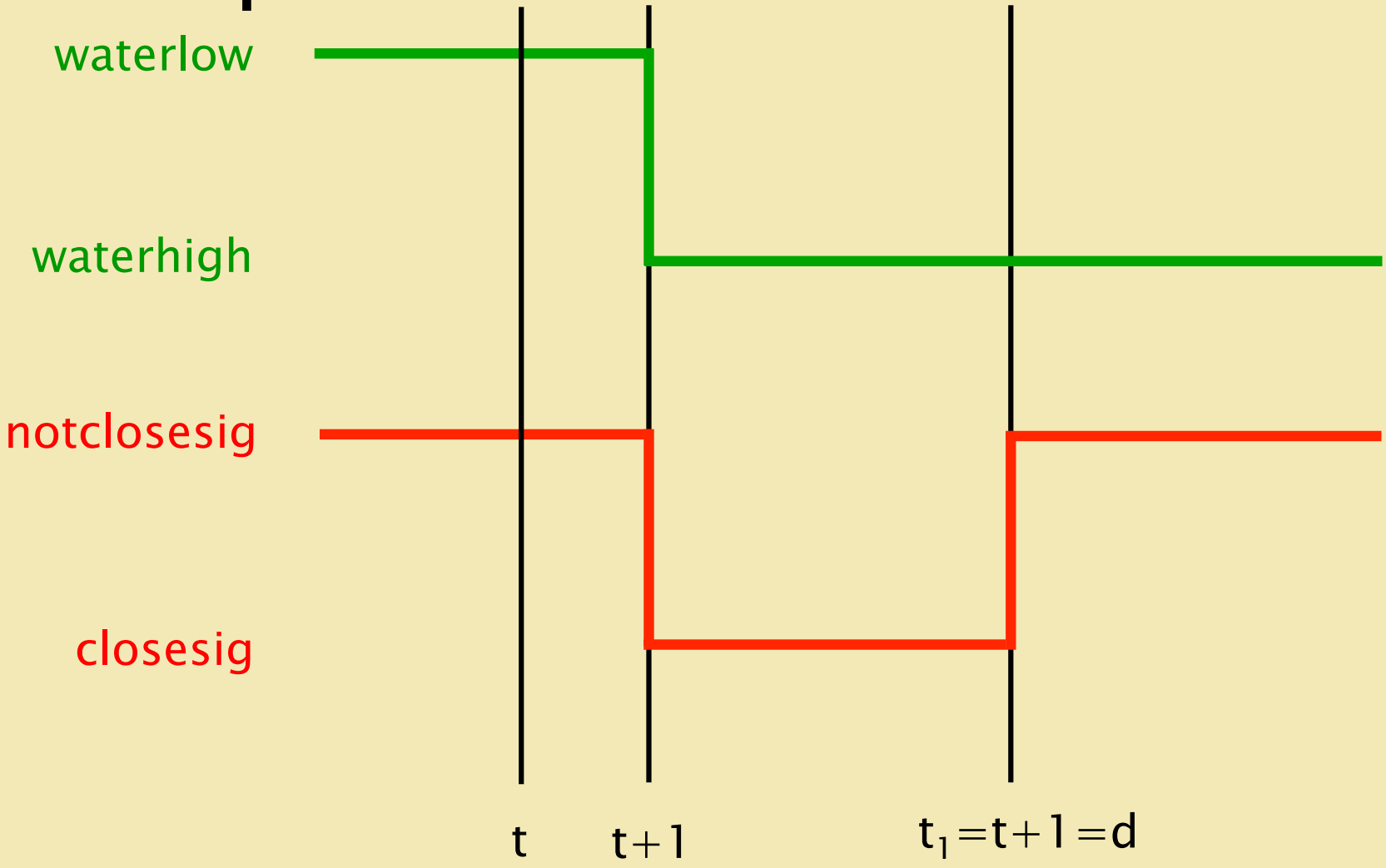
Source: Stephan

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

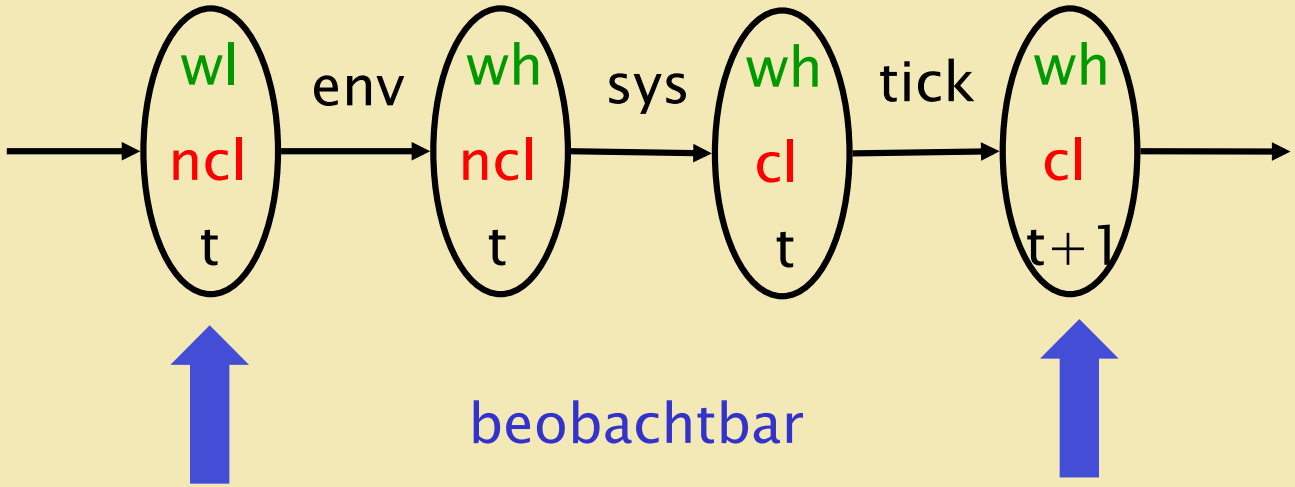
Source: Stephan

Beispiel für einen zeitlichen Ablauf



Source: Stephan

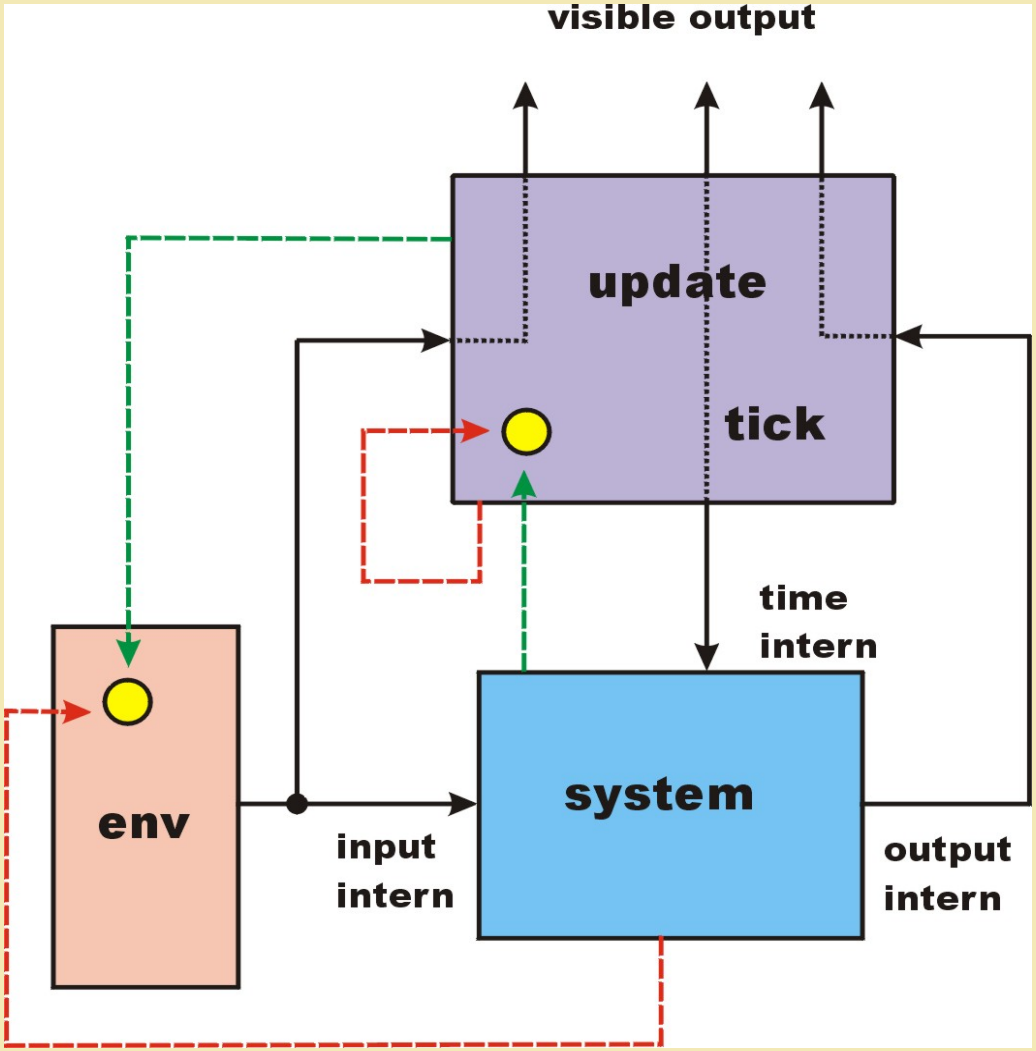
Berechnungsfolgen (Traces)



Verschränkte Ausführung von System – Umgebung – Uhr

Source: Stephan

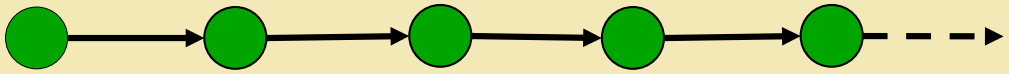
Struktur des NSS-Modells



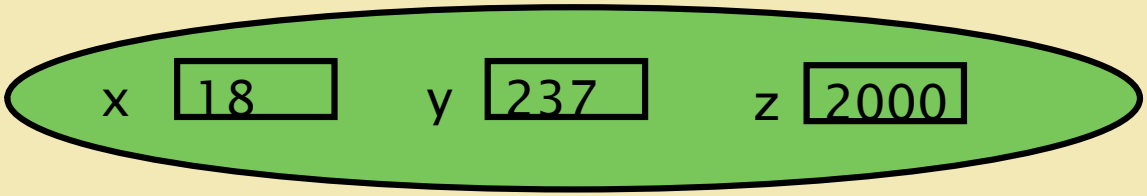
Source: Stephan

Temporallogische Formalisierung

- Beschreibung von Ausführungssequenzen:



- Zustände



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

Source: Stephan

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

Source: Stephan

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

Source: Stephan

Ergebnisse

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

Source: Stephan

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

Source: Stephan

Robertino Control Software

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

Source: Stephan

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

Source: Stephan

Ergebnisse

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

Source: Stephan