Modellierung verteilter Systeme

Grundlagen der Programm und Systementwicklung

Sommersemester 2012

Prof. Dr. Dr. h.c. Manfred Broy

Unter Mitarbeit von Dr. M. Spichkova, J. Mund, P. Neubeck

Lehrstuhl Software & Systems Engineering

Verfeinerung von Systemen

Schrittweise Verfeinerung

Verfeinerung (engl. Refinement) adressiert die Beziehung zwischen Systemmodellen, wie sie im Verlauf einer Entwicklung mit immer mehr Details entstehen

- Formal ist Verfeinerung eine Relation zwischen Systemen (genauer zwischen dem Systemverhalten)
- System(spezifikation) B ist Verfeinerung von System(spezifikation) A, wenn B alle geforderten Eigenschaften von A erfüllt (gegebenenfalls aber noch weitere aufweist):
  - B impliziert alle Eigenschaften von A
  - A kann durch B ersetzt werden
- Es besteht eine enger Bezug zum Begriff **Kompatibilität**

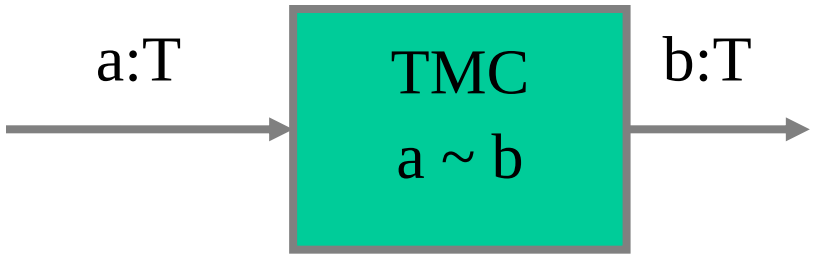
Arten der Verfeinerung

- Eigenschaftsverfeinerung: Verfeinerung der Anforderungen
  - Hinzufügen weiterer Eigenschaften
  - Einbettung in umfangreichere syntaktische Schnittstelle
- Repräsentationsverfeinerung:
  - Verfeinerung der Interaktionsgranularität
  - Änderung der Darstellung von Nachrichten
  - Ersetzen einer Nachricht durch mehrere Nachrichten
  - Änderung vom Kanalnamen
  - Ersetzung eines Kanals durch mehrere Kanäle

Beispiel: Verfeinerung der Spezifikation eines Systems

Übertragungskomponente TMC

TMC
in a: T **out** b: T
a ~ b



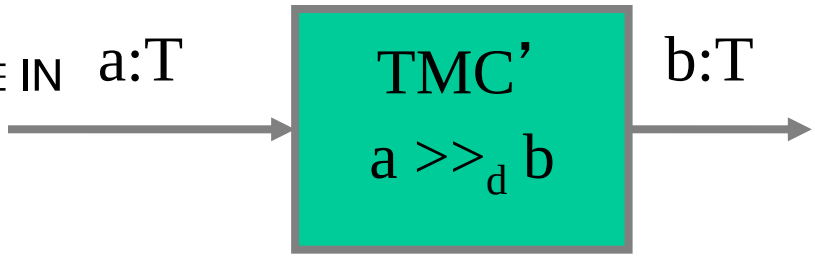
$$a \sim b \equiv (\forall\, m \in T: \{m\}\#a = \{m\}\#b)$$

{m}#a bezeichnet die Anzahl der Vorkommnisse von m im Strom a

Die Zusicherung a ~ b heißt Schnittstellenzusicherung

Beispiel: Verfeinerung der Spezifikation eines Systems

Übertragungskomponente TMC': Gegeben d ∈ IN



TMC'
in a: T **out** b: T
a >>$_d$ b

$a >>_d b \equiv (\forall m \in T, t \in IN: \{m\}\#a\downarrow t \leq \{m\}\#b\downarrow t+d \leq \{m\}\#a\downarrow t+d)$

{m}#a bezeichnet die Anzahl der Vorkommnisse von m im Strom a

$a\downarrow t$ bezeichnet die ersten t Glieder im Strom

TMC' verfeinert TMC

Eigenschaftsverfeinerung

Eine Komponente mit dem Verhalten

$$F : \vec{I} \rightarrow \wp(\vec{O})$$

wird durch eine Komponente mit dem Verhalten

$$F' : \vec{I} \rightarrow \wp(\vec{O})$$

verfeinert, falls für alle Eingaben

$$x \; \hat{I} \; \vec{I}$$

folgende Beziehung gilt:

$$F'(x) \subseteq F(x)$$

Wir schreiben dann

$$F \approx> F'$$

Beispiel der Eigenschaftsverfeinerung: Datenspeicher

Eingabekanal x der Sorte Data | {req}

Ausgabekanal y der Sorte Data

- Forderung für die Speicherfunktion f (für alle $d \in Data$)

  $y = f(x) \wedge d \in y \Rightarrow d \in x$

- Filterfunktion ©

  $\#y = min(\# \{req\} © x, \# Data © x)$

- Für jedes Datenelement sollen höchstens so viele Kopien ausgegeben werden, wie eingegeben und angefordert werden:

  $\# \{d\} © y \leq \# \{d\} © x$

Verfeinerung von Schnittstellenzusicherungen

- Eine Spezifikation mit Schnittstellenzusicherung A‘ verfeinert eine Spezifikation mit Schnittstellenzusicherung A, falls

$$A' \Rightarrow A$$

- Achtung:
  - Die Schnittstellenzusicherung *false* verfeinert damit jede andere Schnittstellenzusicherung
  - Eine Spezifikation mit Schnittstellenzusicherung *false* ist eine widersprüchliche Spezifikation, die nicht implementiert werden kann

Teilfunktionalität

Eingabekanäle I, I'

Ausgabekanäle O, O'

$$\mathrm{I} \subseteq \mathrm{I}' \wedge \mathrm{O} \subseteq \mathrm{O}'$$

Verhalten:

$$f : \vec{I} \rightarrow \vec{O}$$

$$f' : \vec{I}' \rightarrow \vec{O}'$$

f ist eine **Teilfunktion** von f′, falls für alle $x \in \vec{\mathrm{I}'}$ gilt:

$$\mathrm{f}(x \mid I) = \mathrm{f}'(x) \mid O$$

Interaktionsverfeinerung

$$f : (I \to M^\omega) \to \wp(O \to M^\omega),$$
$$\hat{f} : (\hat{I} \to M^\omega) \to \wp(\hat{O} \to M^\omega).$$

Interaktionsverfeinerung: ein Paar von Abbildungen

$$A : (\hat{Z} \to M^\omega) \to \wp(Z \to M^\omega)$$ **Abstraktionsfunktion**

$$R : (Z \to M^\omega) \to \wp(\hat{Z} \to M^\omega)$$ **Repräsentationsfunktion**

für die gilt:

$$R \circ A = Id$$

wobei

$$(R \circ A)(x) = \{z \in A(y) : y \in R(x)\} \qquad Id(x) = \{x\}$$

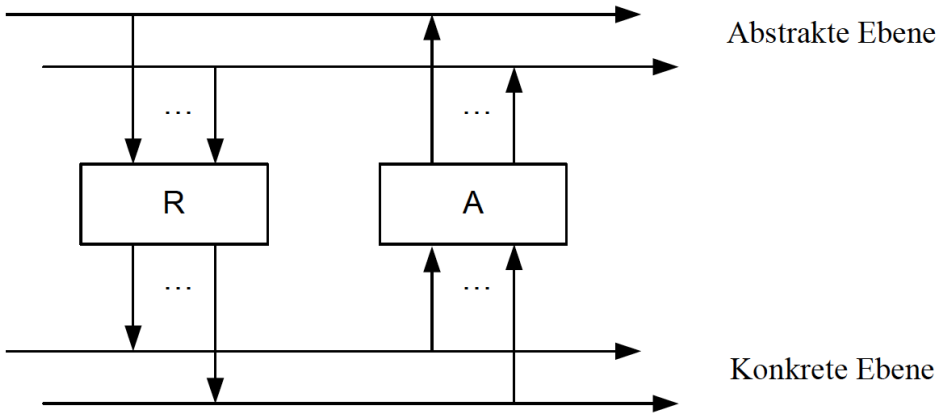
Interaktionsverfeinerung (2)

Änderungen durch Interaktionsverfeinerungen bei einer Komponente:

- Sorte, Zahl und Namen ihrer Eingabe- und Ausgabekanäle,
- Granularität der auf den Kanälen übermittelten Nachrichten

Beschreibung einer Interaktionsverfeinerung:

$$A : \vec{C'} \rightarrow \wp(\vec{C}) \qquad R : \vec{C} \rightarrow \wp(\vec{C'})$$



Interaktionsverfeinerung (U^{-1}-Simulation)

Verfeinerungspaare:

$$A_I : \vec{I}_2 \rightarrow \wp(\vec{I}_1) \qquad R_I : \vec{I}_1 \rightarrow \wp(\vec{I}_2)$$
$$A_O : \vec{O}_2 \rightarrow \wp(\vec{O}_1) \qquad R_O : \vec{O}_1 \rightarrow \wp(\vec{O}_2)$$

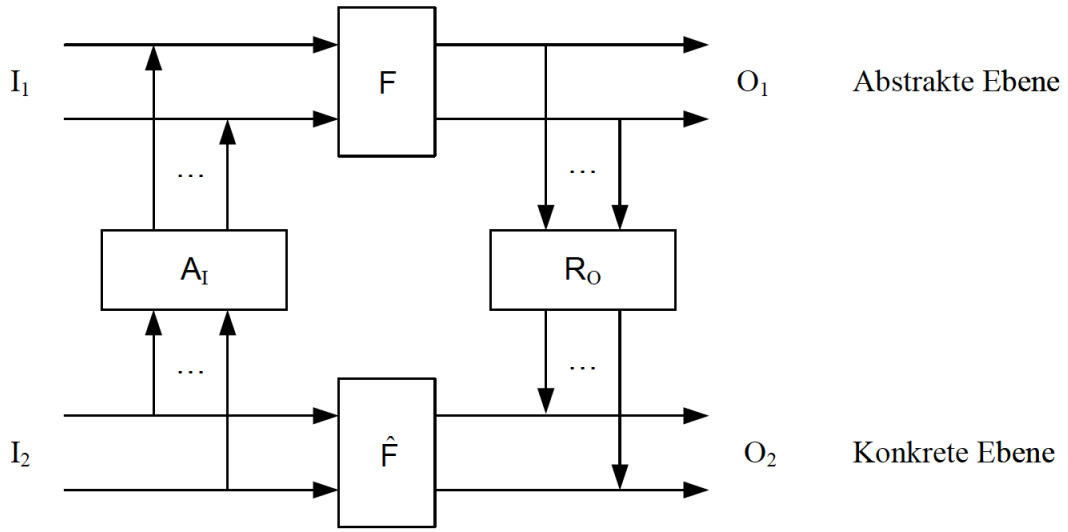
$$\hat{F} : \vec{I}_2 \rightarrow \wp(\vec{O}_2)$$

ist eine Interaktionsverfeinerung des Verhaltens

$$F : \vec{I}_1 \rightarrow \wp(\vec{O}_1)$$

falls der folgende Satz (U^{-1}-Simulation) gilt:

$$\hat{F} \subseteq A_I \circ F \circ R_O$$



Schnittstellenverfeinerung: Interaktionsverfeinerung

Wir brauchen zwei Interaktionsverfeinerungen

$$A_I : (\hat{I} \to M^\omega) \to \wp(I \to M^\omega),$$
$$A_O : (\hat{O} \to M^\omega) \to \wp(O \to M^\omega),$$
$$R_I : (I \to M^\omega) \to \wp(\hat{I} \to M^\omega),$$
$$R_O : (O \to M^\omega) \to \wp(\hat{O} \to M^\omega),$$

so dass gilt (U-Simulation)

$$R_I \circ \hat{f} \circ A_O \subseteq f$$

Interaktionsverfeinerung: 4 Formen

- U^{-1}-Simulation (restriktivster, “stärkster” Begriff)

$$\hat{F} \subseteq A_I \circ F \circ R_O$$

- Abwärtssimulation

$$R_I \circ \hat{F} \subseteq F \circ R_O$$

- Aufwärtssimulation

$$\hat{F} \circ A_O \subseteq A_I \circ F$$

- U-Simulation

$$R_I \circ \hat{F} \circ A_O \subseteq F$$

Beispiel: Parallele und Serielle Subtraktion

Übersetzung:

ein Paar von Zahlenströmen in einem Bitstrom mit Trennzeichen √

$$rep(n \,\&\, a, m \,\&\, b) = bitseq(n) \,\hat{}\, \langle \surd \rangle \,\hat{}\, bitseq(m) \,\hat{}\, \langle \surd \rangle \,\hat{}\, rep(a, b)$$

$$rep(\langle\rangle, b) = rep(a, \langle\rangle) = \langle\rangle$$

Allgemeines zu Verfeinerungen

- Das Konzept der Verfeinerung gibt es in einer Reihe von Ausprägungen
  - Verfeinerung von Datenstrukturen
  - Verfeinerung von Programmen
  - Verfeinerung von Systemen
    - Schnittstellen
    - Architekturen
    - Zustandsmaschinen

Kompatibilität

Grundidee Kompatibilität

- Ein Teilsystem (Komponente) B heißt (ersetzungs-)kompatibel zu einer Komponente A,
  - wenn in beliebigen Systemen S, die A enthalten, folgendes gilt:
  - wenn A in S durch B ersetzt wird, entsteht ein System S‘ das das gleiche (Schnittstellen-)Verhalten wie S zeigt

Formalisiert

- Eine Schnittstellenspezifikation B ist (ersetzungs-)kompatibel zu einer Komponente A, falls B Verfeinerung von A ist
  - es gilt:

$$S = C \otimes A \wedge S' = C \otimes B \wedge A = B \Rightarrow S = S'$$

und

$$S = C \otimes A \wedge S' = C \otimes B \wedge A \approx> B \Rightarrow S \approx> S'$$

Modularität der Verfeinerung

Grundidee Modularität

- Eine Verfeinerung heißt modular,
  - wenn für jede Komponente A mit Verfeinerung durch eine (Komponente) B gilt:
  - in beliebigen Systemen S, die A enthalten, gilt:
  - wenn A in S durch B ersetzt wird, entsteht ein System S‘ das eine Verfeinerung von S ist

Formalisiert

- Eine Verfeinerungsrelation $\approx>$ heißt modular, wenn für alle Systeme S, A, B, C gilt:

$$S = C \otimes A \wedge S' = C \otimes B \wedge A \approx> B \Rightarrow S \approx> S'$$

Repräsentationsverfeinerung: Zustandssysteme

Zustandsübergangssysteme:

- Unmarkiert
- Nichtdeterministisch

Sicherheitseigenschaften: Systeminvarianten

Analyse der Sicherheitseigenschaften von Systemen:

- Invarianten beweisen (Verifikation)
- Gegenbeispiele für Invarianten finden (Testen)

Repräsentationsverfeinerung: Zustandssysteme (2)

- Abstrakter Zustandsraum Σ, konkreter Zustandsraum Σ'

  $\Sigma_0 \subseteq \Sigma, \quad \Sigma'_0 \subseteq \Sigma'$

- Zustandsübergangsfunktionen

  $$\Delta : \Sigma \rightarrow \wp(\Sigma)$$
  $$\Delta' : \Sigma' \rightarrow \wp(\Sigma')$$

- System (Δ, Σ_0) - Abstraktion des Systems (Δ', Σ'_0)

- Abstraktionsfunktion

  $$\alpha : \Sigma' \rightarrow \Sigma$$
  $$\alpha(\Delta'(\sigma)) \subseteq \Delta(\alpha(\sigma))$$

  $$\sigma \in \Sigma'_0 \Rightarrow \alpha(\sigma) \in \Sigma_0$$
  $$\sigma_2 \in \Delta'(\sigma_1) \Rightarrow \alpha(\sigma_2) \in \Delta(\alpha(\sigma_1))$$

Repräsentationsverfeinerung: Zustandssysteme (3)

Sei $I \subseteq \Sigma$ eine Zustandszusicherung auf dem Zustandsraum Σ und $I' \subseteq \Sigma'$ eine Zustandszusicherung auf dem Zustandsraum Σ'

- Koppelrelation:

$$\alpha(\sigma) \in I \Leftrightarrow \sigma \in I'$$

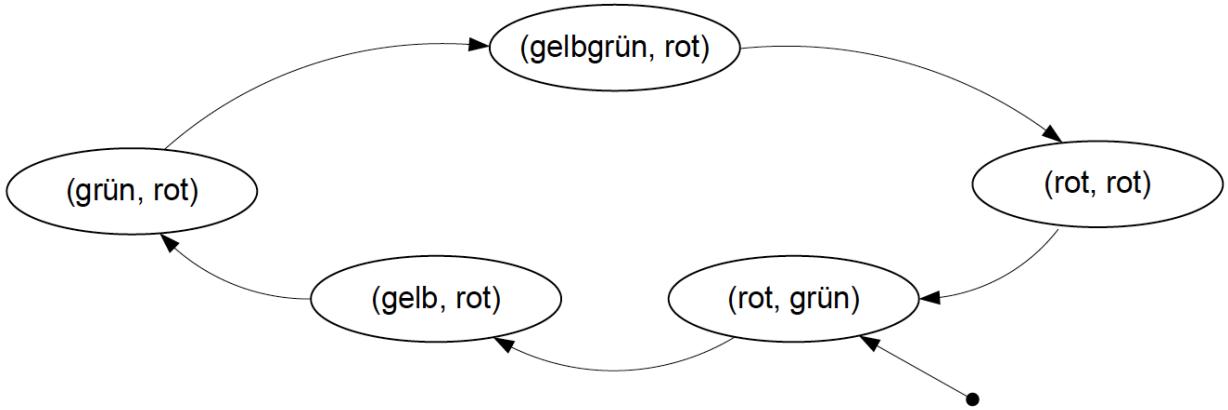
Beispiel: Abstraktion eines Automaten

Kstate = AutoAmpel × FußgängerAmpel

AutoAmpel = { grün, gelb, rot, gelbgrün }

FußgängerAmpel = { grün, rot }

Übergangsdiagramm Δ_K

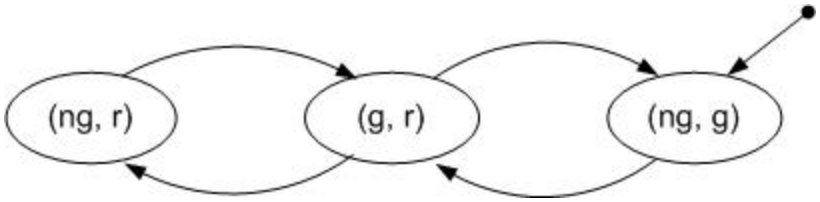


Abstrakter Automat: AState = A × F

A = { g, ng } F = { g, r }

Beispiel: Abstraktion eines Automaten (2)

Übergangsdiagramm Δ_A



α : KState $\longrightarrow$ AState

(a,b)	grün	rot
grün	(g, g)	(g, r)
gelb	(ng, g)	(ng, r)
rot	(ng, g)	(ng, r)
gelbgrün	(ng, g)	(ng, r)

Wir erhalten:

$$y \in \Delta_K(x) \Rightarrow \alpha(y) \in \Delta_A(\alpha(x))$$

und die gekoppelten Invarianten

$$I_K = \{(a,b) \mid \neg(a = \text{grün} \ \wedge \ b = \text{grün})\}$$
$$I_A = \{(a,b) \mid \neg(a = g \ \wedge \ b = g)\}$$

Konstruktion abstrakter Zustandsmaschinen

- Abstrakter Zustandsraum Σ, konkreter Zustandsraum Σ'
- Zustandsübergangsfunktion

$$\Delta' : \Sigma' \rightarrow \wp(\Sigma')$$

- Abstraktionsfunktion

$$\alpha : \Sigma' \rightarrow \Sigma$$

- Konstruierte Übergangsfunktion

$$\Delta : \Sigma \rightarrow \wp(\Sigma)$$

$$\Delta(\sigma) = \{\alpha(\sigma'') : \exists \sigma' \in \Sigma' : \sigma'' \in \Delta'(\sigma') \wedge \sigma = \alpha(\sigma')\}$$

- Invariante für die Maschine Δ: $\boxed{\mathrm{I} = \{\alpha(\sigma') : \sigma' \in \mathrm{I}'\}}$

Abschließende Bemerkungen

Das Konzept der Verfeinerung bildet die Grundlage

- für die Beziehung zwischen Systemmodellen in Entwicklungsschritten
- für Kompatibilität
- für Entwicklung nach „Divide and Conquer“ soweit Verfeinerung modular ist