

kes 20.6.12

7. Softwaretechnische Aspekte

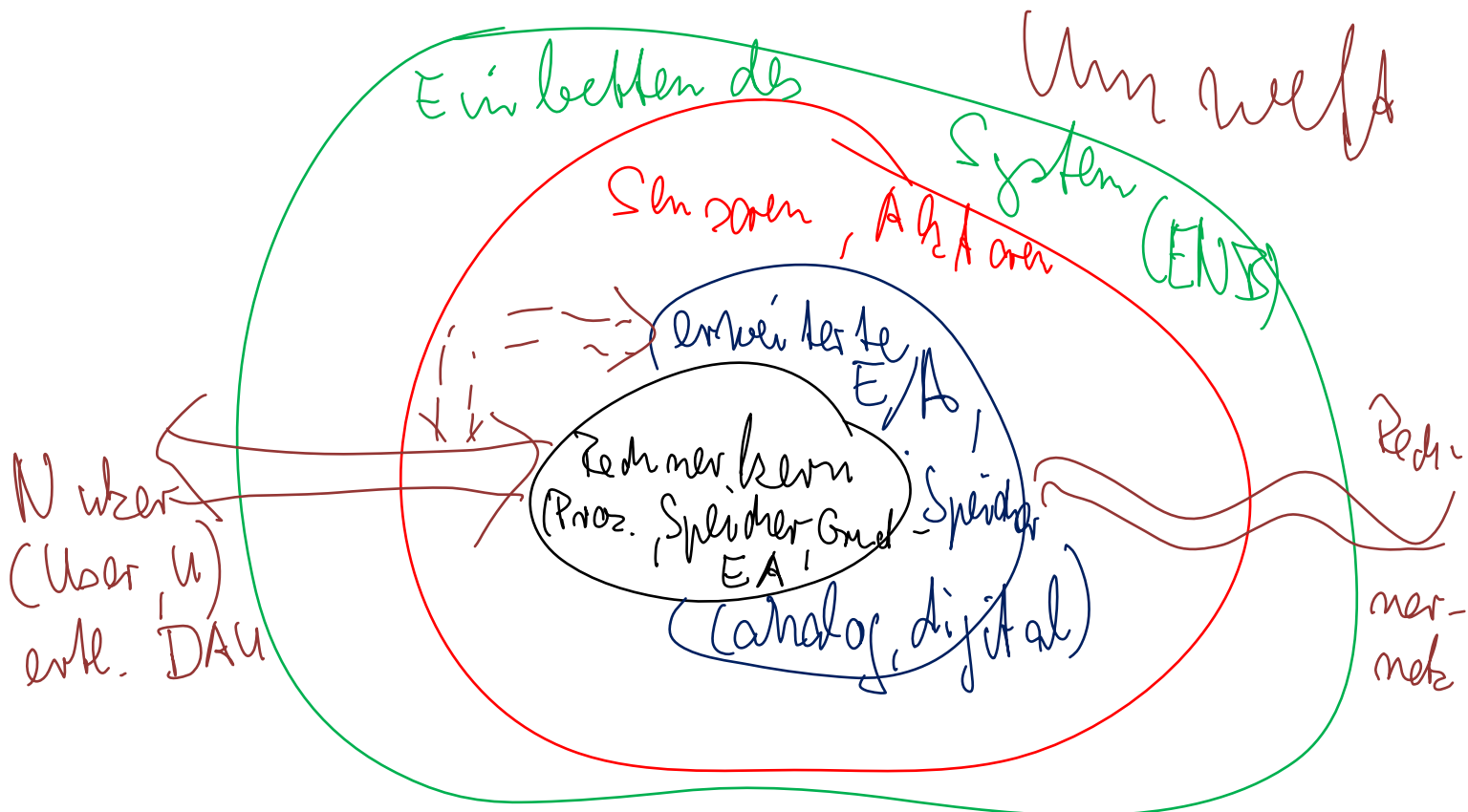
Softwaretechnische Aspekte

7.1. Vorgehensmodellen

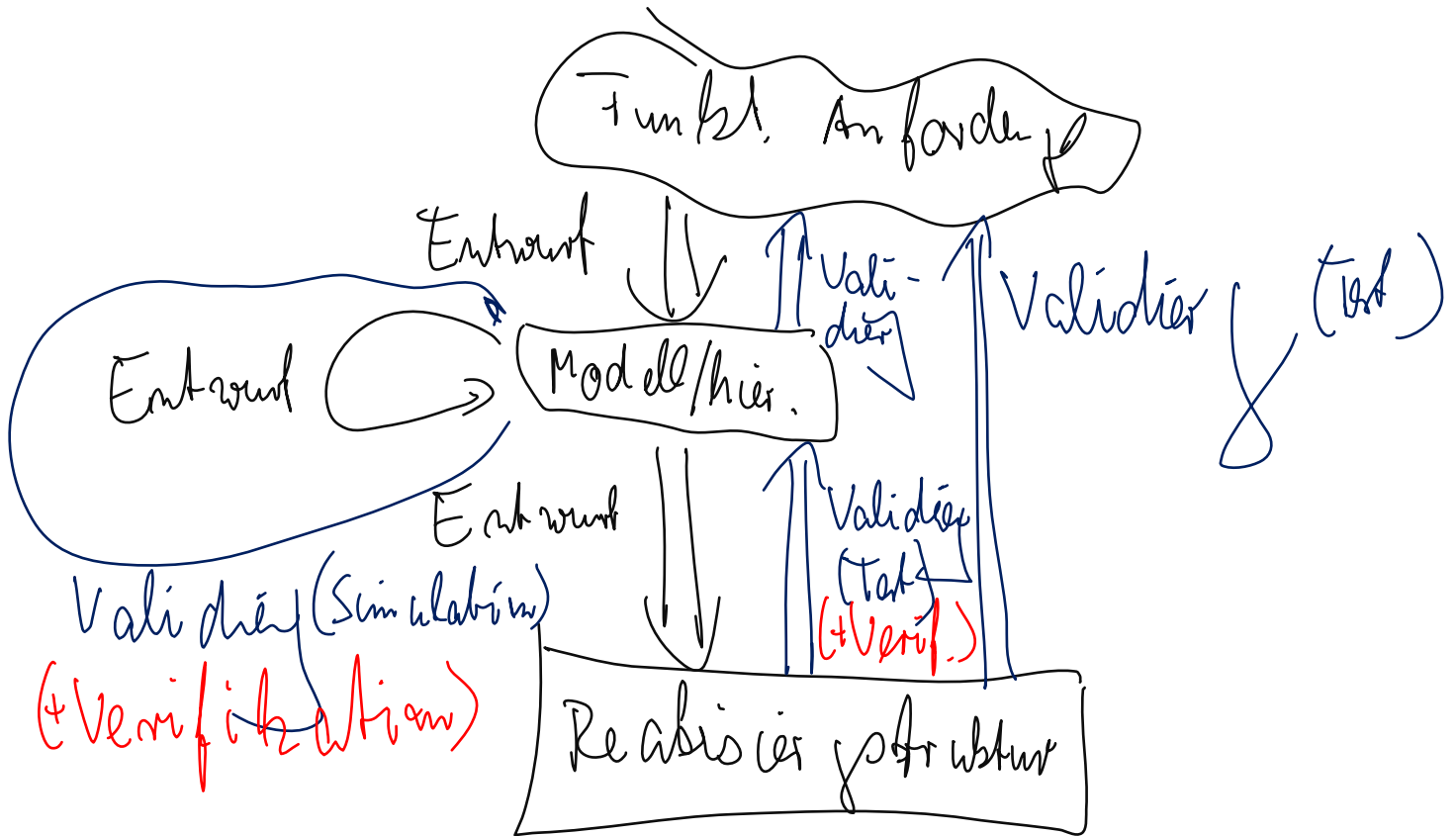
worum: Softwareerstellung (Entwurf, Entwicklung) für Anwendungen in ES (Eingebetteten Systemen)

Entwurf: Überführung von funktionellen Anforderungen in eine Realisierungsstruktur

- lauffähiges Programm auf einer Eingebettete Plattform (Hardware, die speziell für das ES angepasst oder entwickelt wurde)



Eigebettete Plattform: Rechnerkern + erw. EA/Speicher (+ evtl. grundlegende BS-Funktionen)



Validierung: Nachweis, dass ein entworfenes System die funktionellen Anforderungen erfüllt („Baue ich das richtige System?“)

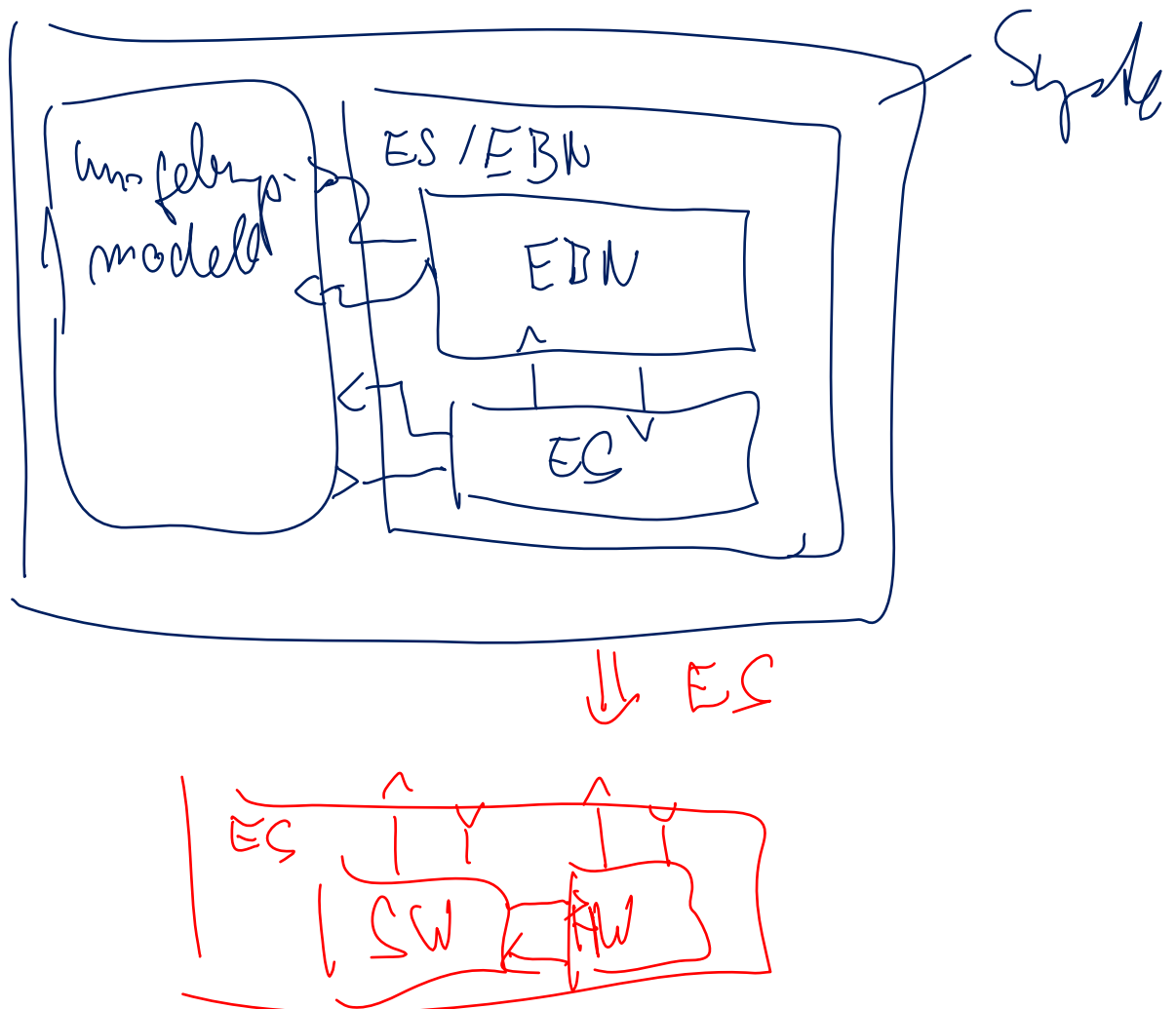
Verifikation: Nachweis, dass das Modell der Abstraktionsebene (i-1) korrekt Eigenschaften der A-Eb. i erfüllt (i.a. formal, -> formale Verifikation)

Def. Vorgehensmodell F7.3

- ➔ Variante für ES: (F7.8)
- ➔ V-Modell ist für ES am verbreitetsten

Grundidee: (mehrere)

- Top-Down-Entwurf
 - Funktionellen Anf. und höheren Abstraktionsebenen (AE)
HW (hardware) und SW (software) gemeinsam
 - auf bestimmter Abstraktionsebene Partitionierung zw. HW und SW
 - danach Einzelentwurf der beiden Teile HW, SW
 - auf den Aufwärtzweig (V links) Integration, auch von HW und Software und dazu speziell Intergrationstest
- ➔ zusätzlich notwendig: Integrationstest ES – ENB (typ. entspricht dem auch eine gemeinsamer Entwurf ES, ENB beim Grobentwurf



7.2. Echtzeit Programmierung

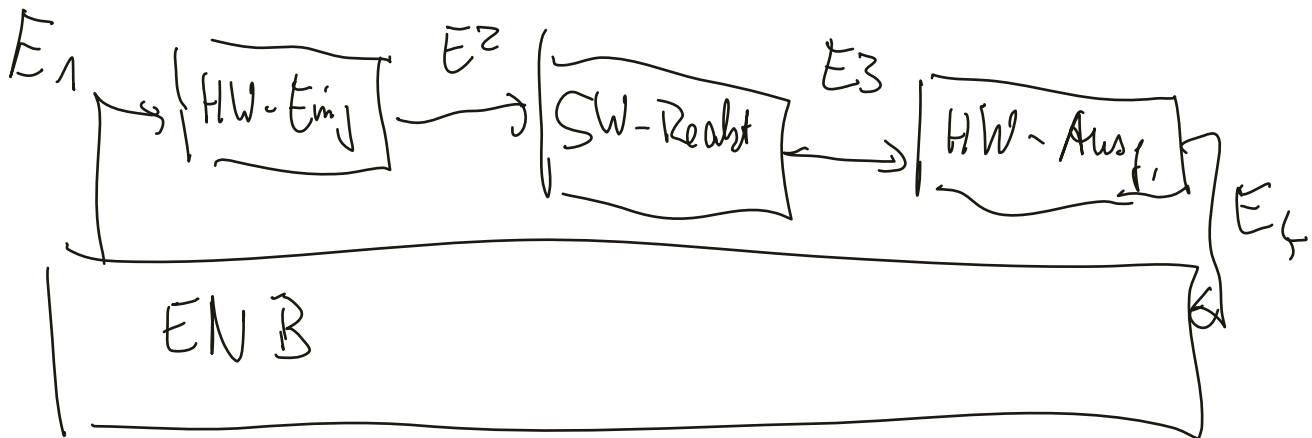
Echtzeit: s.o.

- hier: Unterstützung bei Entwurf, Validierung und Verifikation zur Erzielung bzw. dem Nachweis von Echtzeiteigenschaften

ENB -> erzeugt Echtzeitanforderungen für das ES und damit auch für die SW(ES)

- Betrachtung der gesamten Ereigniskette ENB-HW-SW-HW-ENB

Bsp. rein sequentielle lineare Abhängigkeit:



$$ENB \quad t_{E_4} - t_{E_1} \leq t_{R \max}$$

t_{E_4} : Zeitpunkt der Ausgabe der Reaktion

t_{E_1} : Zeitpunkt der Eingabe des Ereignisses, welches die Reaktion auslösen soll

$t_{R \max}$: maximal zulässige Reaktionszeit

$$tR = (tE2-tE1)+(tE3-tE2)+(tE4-tE3) \\ tHWW+tSW+tHWA$$

- hier (7.2.) Sprachkonstrukte für die Notation von Echtzeiteigenschaften
- Implementierungen aller Sprachelemente unter Berücksichtigung der Echtzeitanforderungen
- evgtl Betriebssystemfunktionen zur Unterstützung der Echtzeiteigenschaften (Implementierung, Gewährleistung)

Sprachen (F7.20)

am häufigsten: C, C++, ia. in Variante für ES („embedded“)

Variante „embedded“:

- Erweiterung um Sprachelemente zur Echtzeit, HW-Zugriff usw.
- Gewährleistung der Echtzeiteigenschaften im Compiler
- Verringerung um Sprachelemente, die die Echtzeit nicht gewährleisten bzw. im Embedded Bereich unnötig sind und evtl. leistungsabhängig nicht realisierbar sind

veränderter Entwicklungsablauf (Werkzeuge): (F 7.21)

- zusätzlich: einfache Einbindung von Assemblerteilen → *Cr. Assembl.*
- „Cross“-Werkzeuge (d.h. Werkzeug läuft nicht auf dem *Linker* Zielprozessor)
- bei ROM-Realisierung (verschiedene ROM-Varianten), (nicht ladbar) notwendig die Anpassung auf physisch feste Adressen
- spezielle Vorkehrungen zum Testen *Remote-Debugger* *Locator* *Programm*