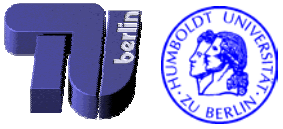


Komponierbarkeit nichtfunktionaler Eigenschaften - Versuch einer Definition

Matthias Werner
Technische Universität Berlin
mwerner@cs.tu-berlin.de

Jan Richling
Humboldt-Universität zu Berlin
richling@informatik.hu-berlin.de

Gliederung

- Motivation
- Eigenschaften in Kompositionen
- Definitionen
- Fallstudie Message Scheduled System (MSS)
- Beweis von Komponierbarkeit
- Zusammenfassung und zukünftige Arbeiten

Einleitung / Motivation

- Problem: Hohe Komplexität verteilter Systeme
  - Erschwert Entwurf und Implementation von Systemen
  - Erschwert Verständnis interner Vorgänge
- Bekannte Lösungen
  - Zerlegung
  - Wiederverwendung
  - Standardschnittstellen
  - Standard-“Teile“ (Komponenten)
- Problem wird in eine Menge von überschaubaren Teilproblemen zerlegt

Einleitung / Motivation

- Modularisierung und Strukturierung:
  betrachten *funktionale* Eigenschaften, da Zerlegung entsprechend der funktionalen Struktur
- In vielen Fällen nicht ausreichend
- *Nichtfunktionale* Eigenschaften sind bedeutsam: zeitliches Verhalten, Ressourcenbedarf, Verläßlichkeitseigenschaften
- Nichtfunktionale Eigenschaften werden bei der Betrachtung des funktionalen Kerns abstrahiert
- Nichtfunktionale Eigenschaften sind meist orthogonal zur funktionalen Zerlegung eines Problems
- Ziel: “Komponierbarkeit”

Komponierbarkeit

- Was ist „Komponierbarkeit“?
- In der Forschung oft benutzt, aber nicht einheitlich definiert:
  - Bergmans:
    Composability allows for the modular specification of modules with multiple independent concerns
  - Kopetz:
    An architecture is said to be composable with respect to a specified property if the system intergration will not invalidate this property, once the property has been established at the subsystem level. [..]
  - Malek:
    A set of elements with given properties is composable iff the calculation of compositions‘ properties needs polynomial time.
- Erforderlich: Allgemeine Definition (domainübergreifend)

Konzepte und Begriffe

- Ziel: Komponierbarkeit als Eigenschaft einer (System)-Architektur
- Architektur: Satz von Regeln zur Erstellung eines Systems
- Grundbausteine: Elemente
  - Elemente können zu Systemen komponiert werden
  - Systeme sind ebenfalls Elemente
  - Kompositionsoperatoren definieren mögliche Kompositionen

Eigenschaften und Qualitäten

- Jedes Element hat Eigenschaften
  - Existenz der Eigenschaft (z.B. Eigenschaft „Farbe“ existiert)
  - Wert der Eigenschaft (z.B. Eigenschaft „Farbe“ hat den Wert „rot“)
  - Jedes Element hat unendlich viele Eigenschaften
- Basisqualitäten (oder Qualitäten): Satz von Eigenschaften, die das Element (für einen gegebenen Zweck) charakterisieren
- Eigenschaften sind aus Basisqualitäten ableitbar
- Variablen repräsentieren Qualitäten eines Elements
- Veränderungen der Variablen repräsentieren Zustandsübergänge: Element kann als State-Machine gesehen werden

Eigenschaften und Komposition

- Komposition von Elementen kann Elementvariablen (Qualitäten und Eigenschaften) verändern:
  - Invariante
    Variable bleibt unverändert und identifizierbar (also an eines der Elemente gebunden)
  - Gebundene Qualität
    Variable wird an einen neuen Wert oder Wertebereich gebunden, bleibt aber einem der Elemente zuordenbar
  - Verschwindende Qualität
    Variable kann im neuen System nicht mehr identifiziert werden
  - Auftauchende Qualität
    Eine neue Variable wird generiert
  - Übertragene Qualität
    Variable wird an andere Variable gebunden, die zum System gehört

Eigenschaften von Architekturen

- Erreichbarkeitseigenschaft:
  - Es gibt ein Element der Architektur, das die gewünschte Eigenschaft hat
- Sicherheitseigenschaft
  - Alle Elemente der Architektur haben die gewünschte Eigenschaft
- Ziel: Komponierbarkeit in Bezug auf Eigenschaften
  - Vorteil: Eigenschaft der Architektur führt zu Eigenschaft ihrer Elemente

Komponierbarkeit in Bezug auf Eigenschaften

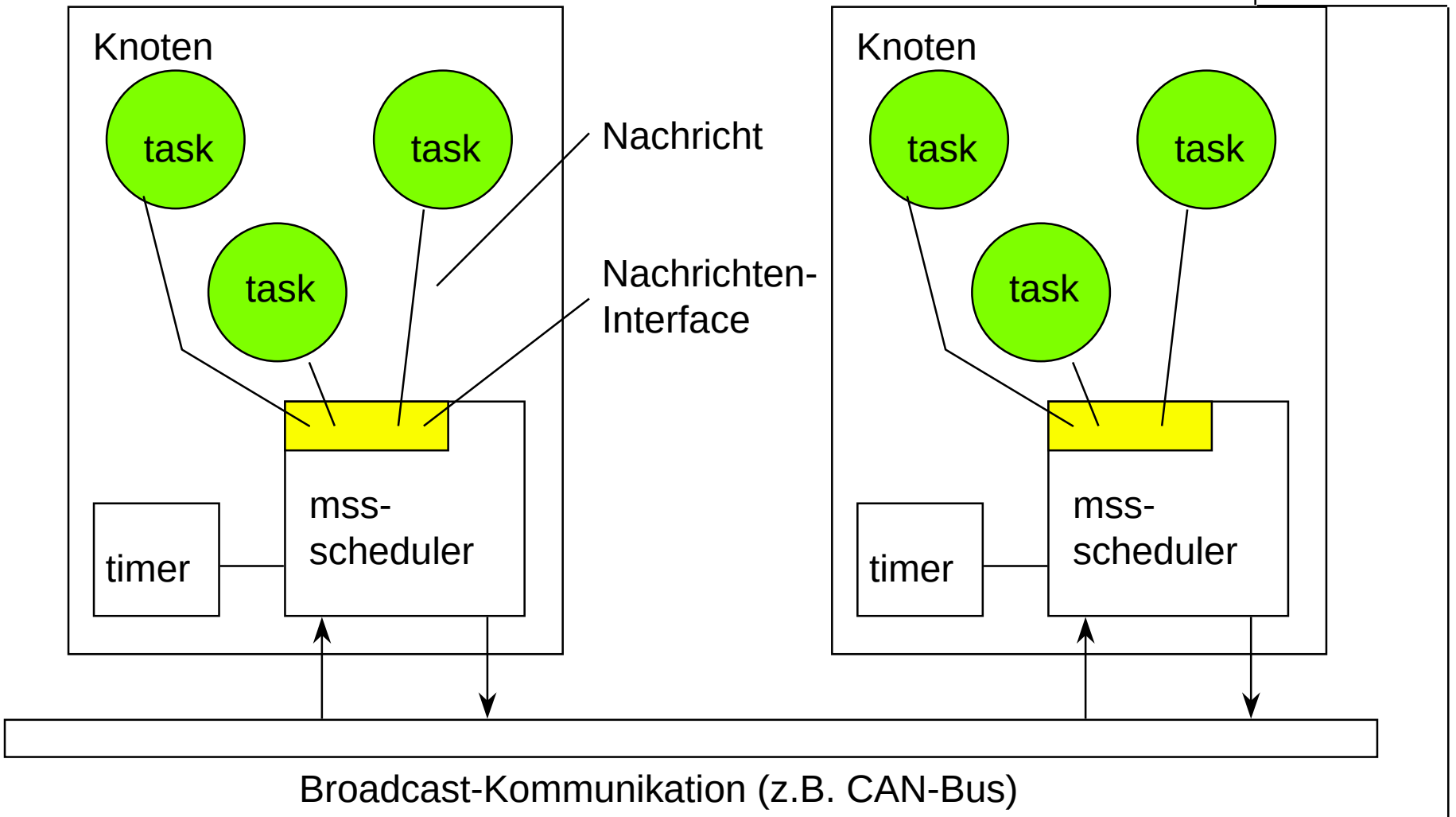
- Eine Systemarchitektur ist komponierbar in Bezug auf eine Sicherheitseigenschaft, wenn sie ausschließlich die Komposition von Systemen erlaubt, die diese Eigenschaft besitzen.

- Eine Systemarchitektur ist komponierbar in Bezug auf eine Erreichbarkeitseigenschaft, wenn sie die Komposition von wenigstens einem System erlaubt, das diese Eigenschaft besitzt.

Fallbeispiel: Message Scheduled System (MSS)

- Ziel:
  - Eingebettete Echtzeitsysteme (Fahrzeuge, Flugzeuge, Automatisierung)
  - Entwicklung einer Architektur mit Unterstützung von Komponierbarkeit in Bezug auf das zeitliche Verhalten (Sicherheitseigenschaft)
    - Garantierte Deadlines für Tasks und Nachrichtenübertragungen
    - Garantierte End-zu-End-Zeiten
- Idee:
  - Globale Verträge mit lokalem Wissen
  - Mehrstufige Abbildung von Komponierbarkeits- auf Schedulingentscheidungen
- Technische Voraussetzungen:
  - Echtzeitfähige Knoten an einem Echtzeitkommunikationsmedium
  - Globale Prioritäten

MSS: Architektur



MSS: Elemente

- Tasks
  - Erzeugen aus einem Satz von periodischen Eingangsnachrichten einen Satz von Ausgangsnachrichten (mit Deadline)
  - Komposition mit: Knoten oder System
- Knoten
  - Führt Tasks aus; verfügt über “Nachrichtenmanager”, der den Nachrichtentraffic der Tasks verwaltet und über die Nutzung des Kommunikationsmediums entscheidet
  - Komposition mit: Task, Kommunikationsmedium oder System
- Kommunikationsmedium
  - Echtzeitfähiger Bus, der die Knoten verbindet
  - Höchstens einmal in jedem System vorhanden
  - Komposition mit: Knoten oder System

MSS: Komponierbarkeit

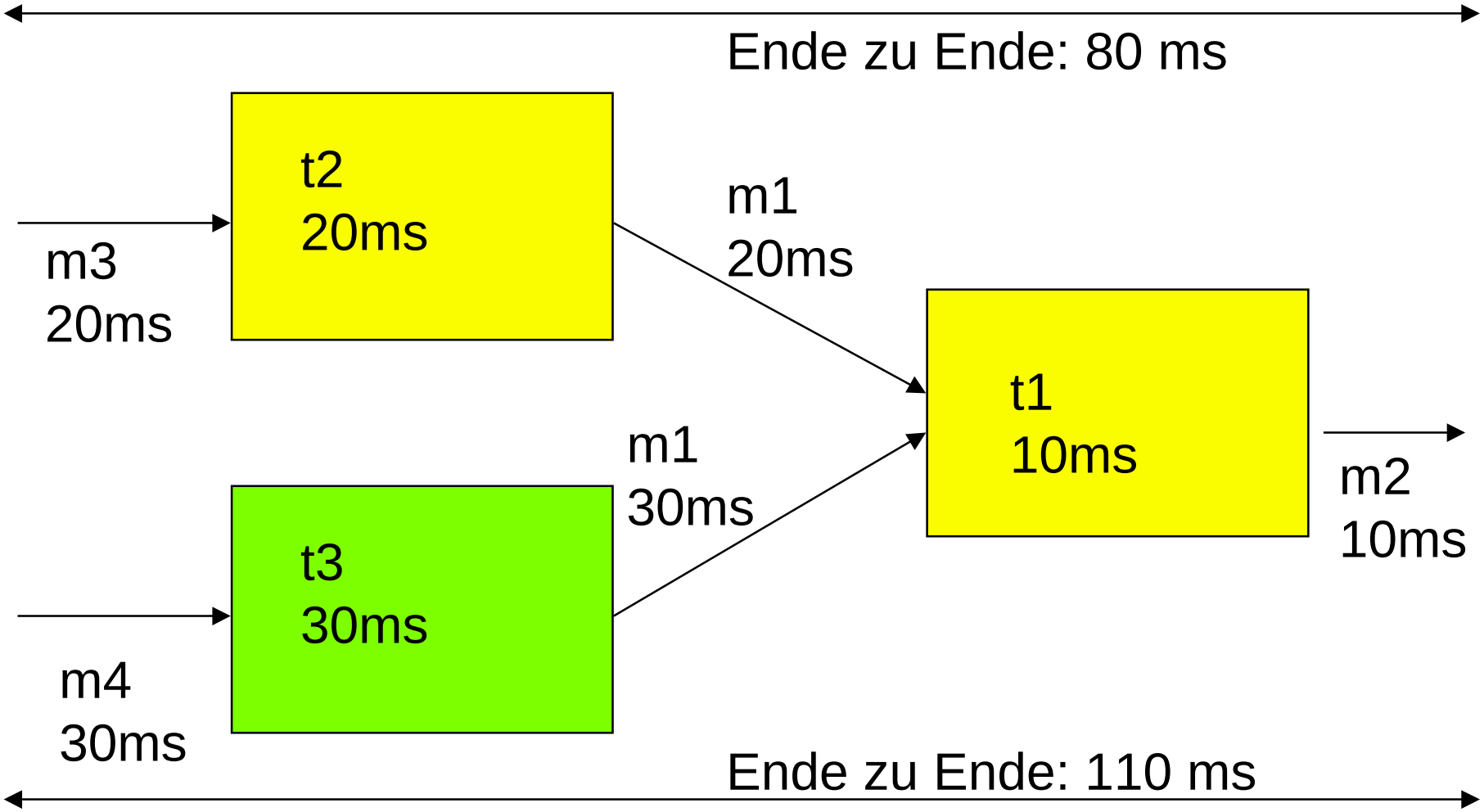
MSS bildet Entscheidungen der Komponierbarkeit auf Schedulingentscheidungen ab:

- Lokales Scheduling aller Tasks auf einem Knoten
  - betrachtet die lokalen Ressourcen eines Knoten wie CPU-Zyklen
- Globales Scheduling auf dem Echtzeit-Netzwerk
  - zwischen allen Nachrichten verschiedener Typen
    - betrachtet die Nachrichten-Slots auf dem Kommunikationsmedium
  - aller Nachrichten gleichen Typs an den gleichen Empfänger
    - betrachtet die Inanspruchnahme eines Empfängers, an den mehrere Sender Nachrichten senden
- Berechnungen sind möglich in linearem Aufwand

MSS: Garantien

- Existenz von Schedules auf allen Ebenen:
  Alle Anforderungen der Komponenten erfüllt
  - Die Berechnung der Schedules ist in linearer Zeit und unter Nutzung bereits existierenden Wissens (Parameter wie Last) möglich
  - Die Berechnungen basieren auf eingeführten Verfahren der Echtzeit-Forschung (RMA, EDF)
  - Bandbreite und Last als Abstraktion
- Systemeigenschaften können aus den Komponenteneigenschaften berechnet werden (beispielsweise End-zu-End-Zeiten)
  - Einfache Berechnung

Komponiertes System

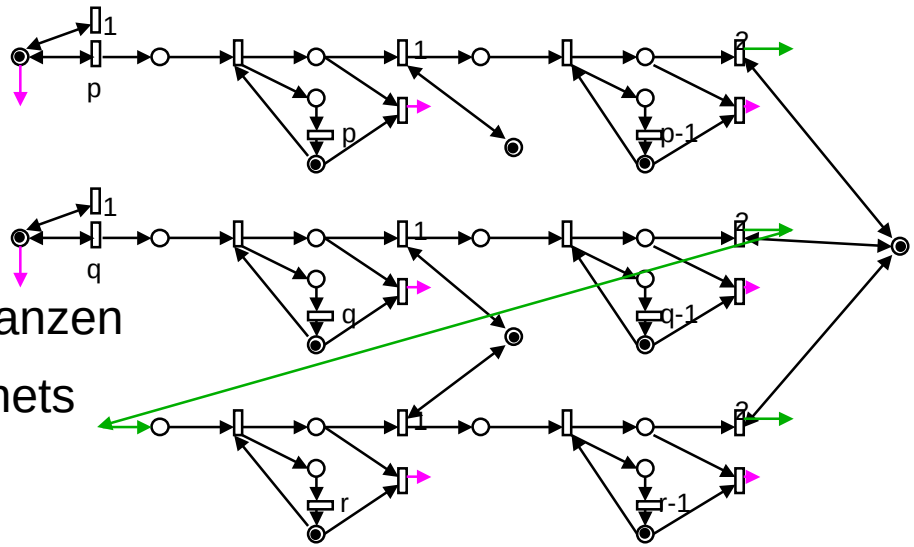


Anwendung des Konzeptes auf MSS

- MSS als Architektur hat die Eigenschaft, sicher komponierbar in Bezug auf die Einhaltung zeitlicher Eigenschaften zu sein:
  - Einhaltung von Taskdeadlines im komponierten System (invariante Eigenschaft)
  - Einhaltung von End-zu-End-Nachrichtenauslieferungszeiten (auftauchende Eigenschaft)
- Es ist nicht möglich, in MSS Systeme zu bauen, die diese Eigenschaft nicht haben
  - Vereinfachtes Design eines Systems
  - Vereinfachte Verifikation von Systemen

Beweis der Komponierbarkeit

- Zu zeigen:
  - Jedes nach den Regeln von MSS komponierte System hat die geforderten Eigenschaften
- Beweisansatz:
  - Beschreibung mit Timed Petrinets
  - Einschränkungen der Architektur führt zu einer eingeschränkten Klasse von Petrinetzen
  - Nichterreichbarkeit von Fehler-zuständen in dieser Klasse zeigen
- Status:
  - Einfache Beweise für konkrete Instanzen
  - Zustandsgleichung für Timed Petrinets entwickelt
  - Allgemeiner Beweis in Arbeit



Zusammenfassung und Ausblick

- Zusammenfassung
  - Domainübergreifendes Konzept für Komponierbarkeit als Eigenschaft einer Architektur
  - Betrachtung von Eigenschaften auf Element- und Systemebene
  - Definition von Komponierbarkeit in Bezug auf Eigenschaften
  - MSS als Beispiel für eine sicher komponierbare Architektur
- Ausblick
  - Einbindung existierender Ansätze in unser Konzept
    - Aspekt-orientierte Programmierung
    - Hardware/Software-Codesign
  - Verifikation sicherer Komponierbarkeit