

# Energieeffiziente Software-Systeme

Christian Bunse – Fachhochschule Stralsund  
Hagen Höpfner – Bauhaus-Universität Weimar

Echtzeit 2010 – Eingebettete Systeme

Boppard, 18.11.2010



**fachhochschule**  
stralsund

university of  
applied  
sciences

fachbereich school of  
elektrotechnik electrical engineering +  
+ informatik computer science

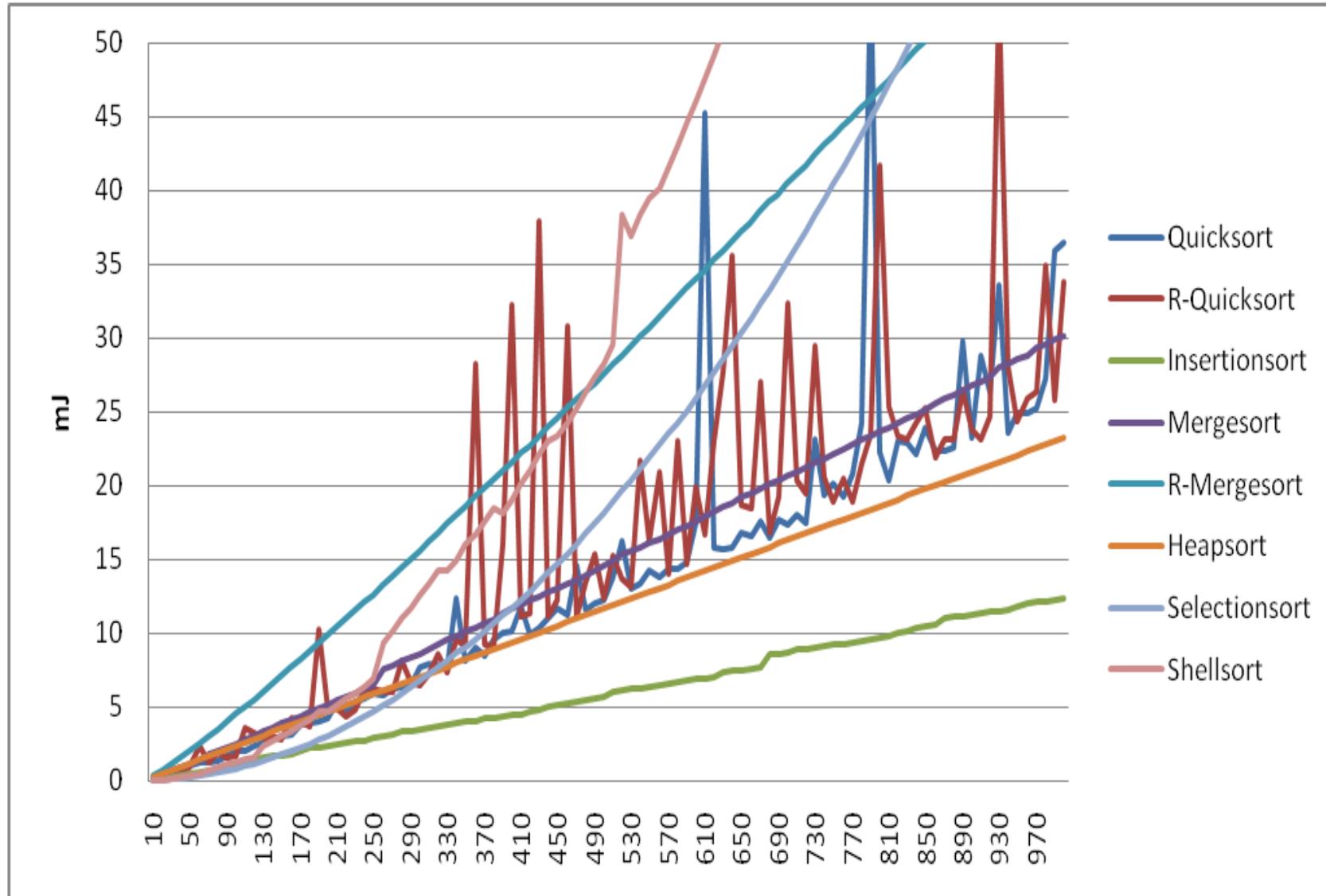
# Energie & Informationstechnologie

- ◆ Ein Standard Computer (PC)
  - Typischer Energieverbrauch: 150 Watt -> 125 KW/h pro Jahr
  - 62% aller Europäischen Haushalte (25 Länder) und 97% aller Firmen nutzen einen Computer
    - ◆ Europa verbraucht mehrere TW/h allein für Computer
    - ◆ Zusätzlich existieren in der EU mehr als 390 Mill. Mobiltelef., ...
  - 2020 werden ca. 35% des globalen Energieverbrauchs für Computer/Computing aufgewandt.
  
- ◆ Eine Google-Suche repräsentiert einen CO<sub>2</sub> Ausstoß von 2-10g [Wissner-Gross]
  
- ◆ Forschung im Bereich Energie-Verbrauch fokussiert häufig auf Hardware Aspekte. Stichworte: Green-IT, Low-Power Design, „Dynamic Voltage Scaling“, ...
  
- ◆ Frage: Hat Software einen Einfluss auf den Energieverbrauch?

# Energie & Software

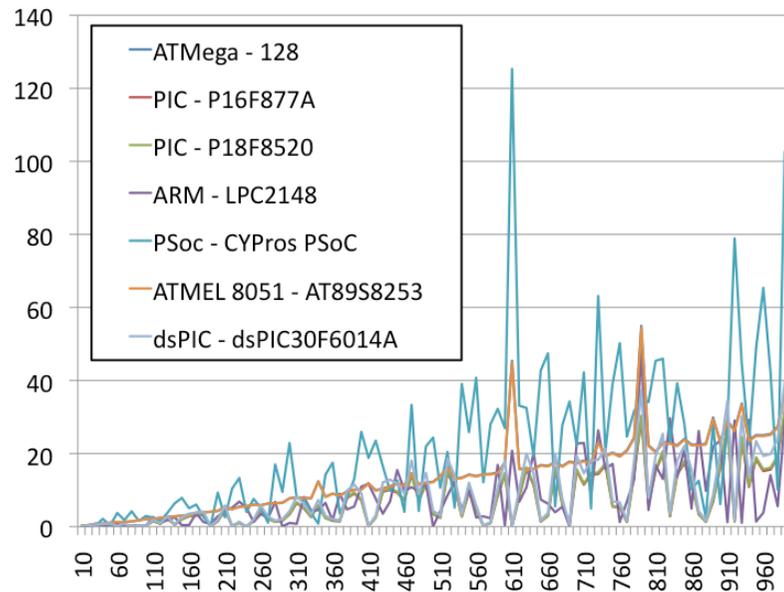
- ◆ Literatur: Software hat einen Einfluss [Huels2002]
  - Austausch „double“ gegen „float“ [Akzeptable Qualität]
  - Energieverbrauch auf 34%, Zyklenanzahl auf 35 % gesenkt
- ◆ Experimente auf Mikrocontrollerbasis:
  - Energieverbrauch wird nicht nur durch Laufzeit/Komplexität definiert
  - Speicherplatz, Rekursion, etc. sind wichtig
- Software-Optimierung zur Senkung des Energieverbrauchs
- ◆ Offene Frage:
  - Welche Faktoren sind von Bedeutung?
    - ◆ Für mobile Systeme ist insbesondere die Kommunikation ausschlaggebend für den Energieverbrauch [Domis, Zhang, ...]
- Software-Optimierung mit Fokus auf Kommunikation

# Energie & Software



# Energie-Signaturen

- ◆ Fallstudien und Messwertanalyse lassen vermuten:
  - Algorithmen haben eine eindeutige Energiesignatur (statistisch signifikante Korrelation)

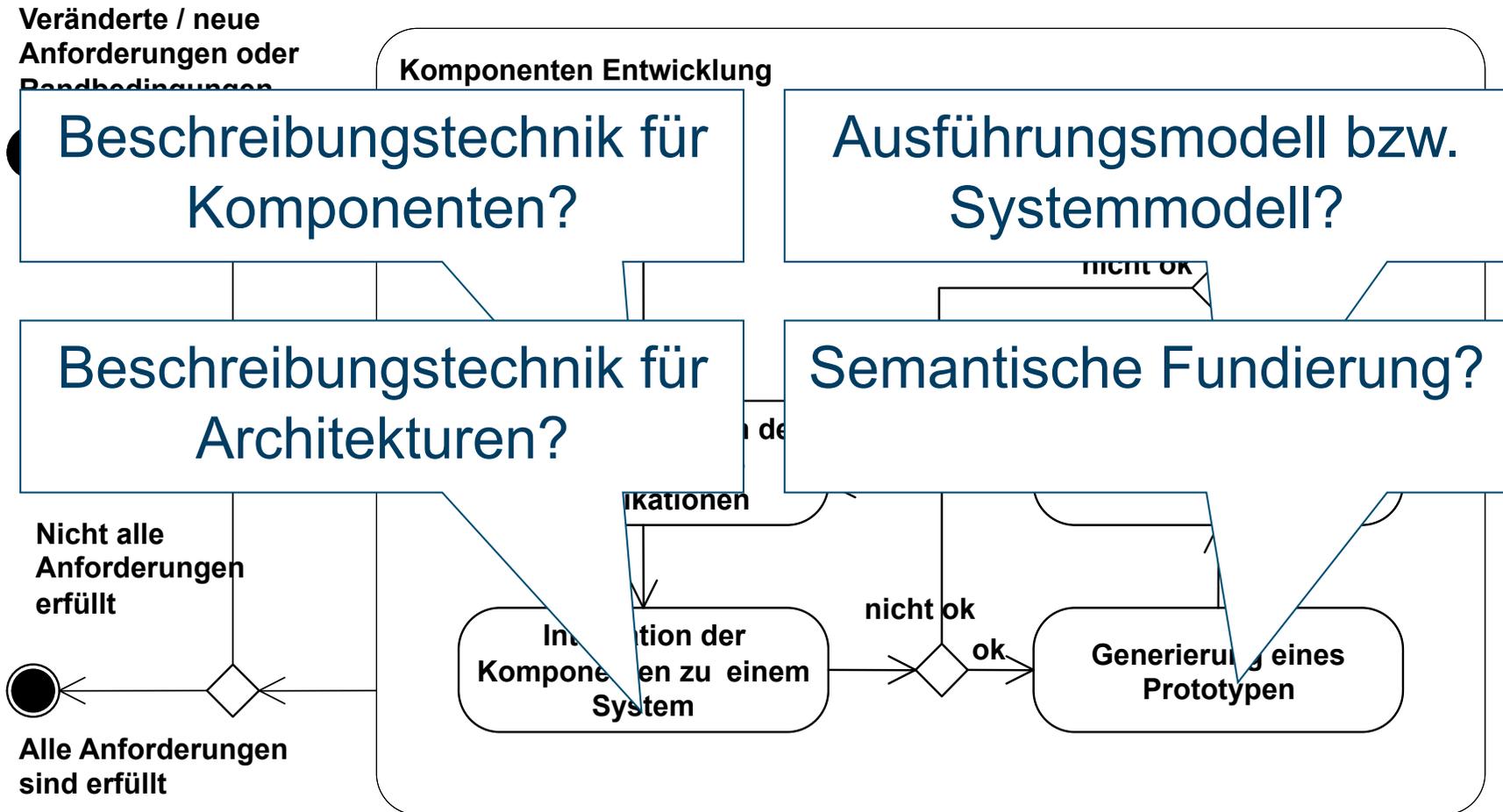


- Charakteristik lässt sich auf zusammengesetzte Systeme übertragen
  - Hypothese: Trifft auch für Komponenten und Systeme zu
  - Frage: Schachtelungstiefe

# Zentrale Frage: Wo im SW-Lebenszyklus hat Optimierung den größten Effekt

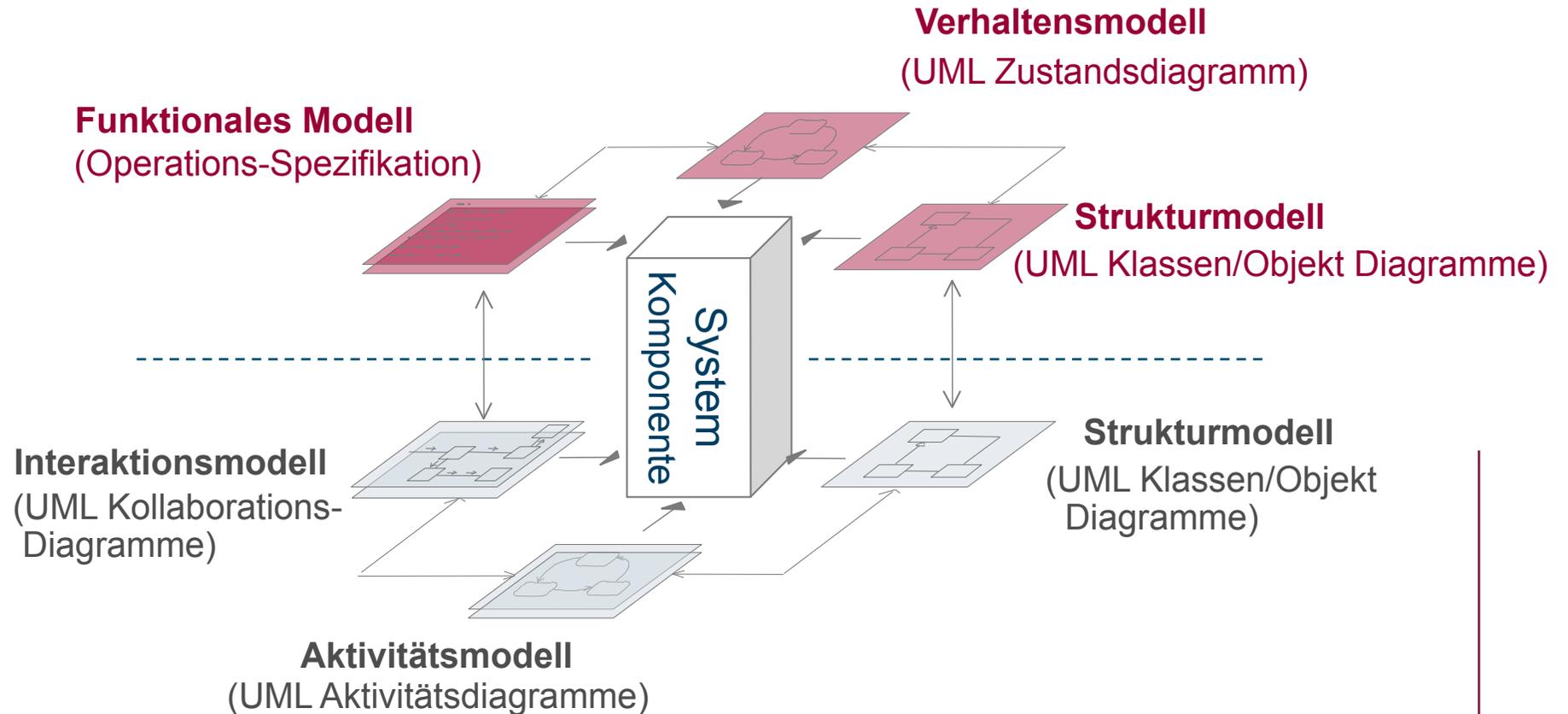
- ◆ Stand der Praxis: Optimierung auf Code- oder Assembler-Ebene
  
- ◆ Änderungskosten steigen mit den Lebenszyklusphasen [Boehm]
  - Optimierung so früh wie möglich
    - ◆ Architektur/Entwurf (genügend Details, Änderungen „einfach“ möglich)
    - ◆ Erfordert systematische Methodik
  
- ◆ Gewählter Ansatz: MARMOT
  - Modellgetrieben
  - Unterstützung von Produktlinien und Komponenten
  - Speziell adaptiert für eingebettete Systeme
  - Realisierung von systematisierte Wiederverwendung (auch auf COTS Ebene)

# „Idealbild“ der Komponenten-basierten Entwicklung



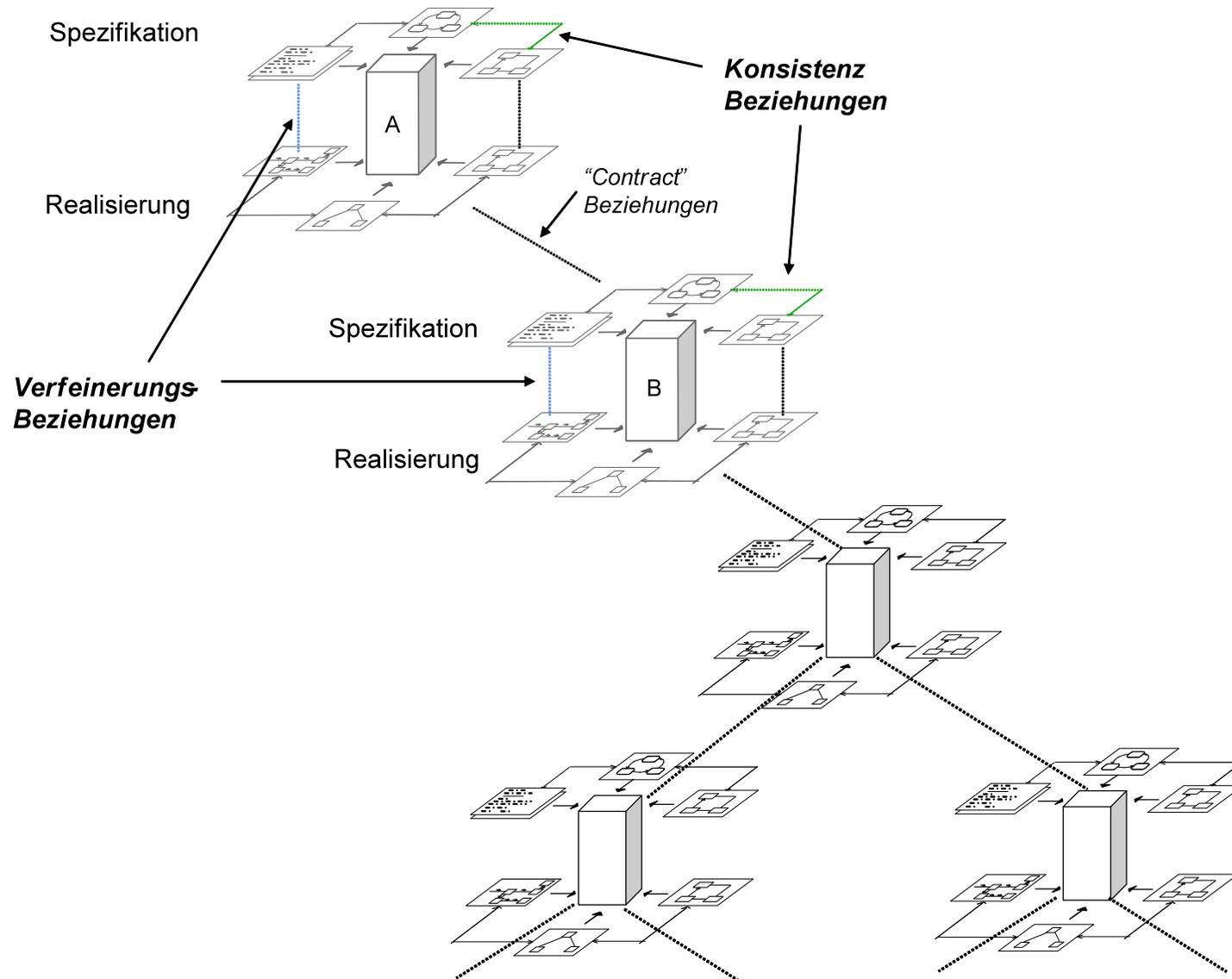
# Modellierung von Komponenten

## Spezifikation



Realisierung

# Komponenten-Hierarchie



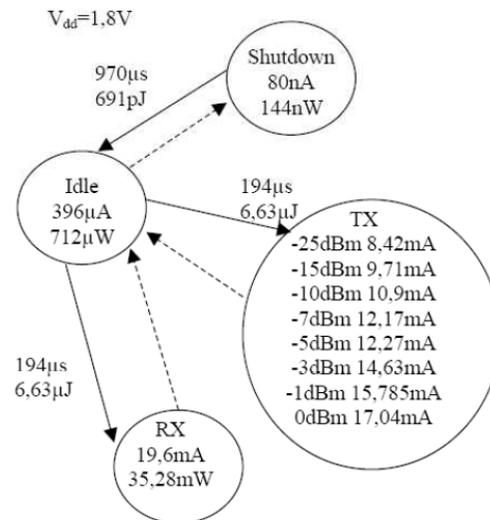


# Empirische Ergebnisse

		Original	ATMega32	PICF	Adapt-	Adapt+	Door
LOC		310	310	320	280	350	490
Model Size (Abs.)	NCM	8	8	8	6	10	10
	NCOM	15	15	15	11	19	29
	ND	46	46	46	33	52	64
Model Size (Rel.)		1	1	1	1	0.8	1
		3.25	3.25	3.25	2.5	3	3.4
		1.375	1.375	1.375	1.33	1.5	1.6
Reuse	Reuse Fraction(%)	0	100	97	100	89	60
	New (%)	100	0	3	0	11	40
	Unchanged (%)	0	95	86	75	90	95
	Changed (%)	0	5	14	5	10	5
	Removed (%)	0	0	0	20	0	40
Effort (h)	Global	26	6	10.5	3	10	24
	Hardware	10	2	4	0.5	2	8
	Requirements	1	0	0	0.5	1	2
	Design	9.5	0.5	1	0.5	5	6
	Implementation	3	1	3	0.5	2	4
	Test	2.5	2.5	2.5	1	2	4
Quality	Defect Density	9	0	2	0	3	4

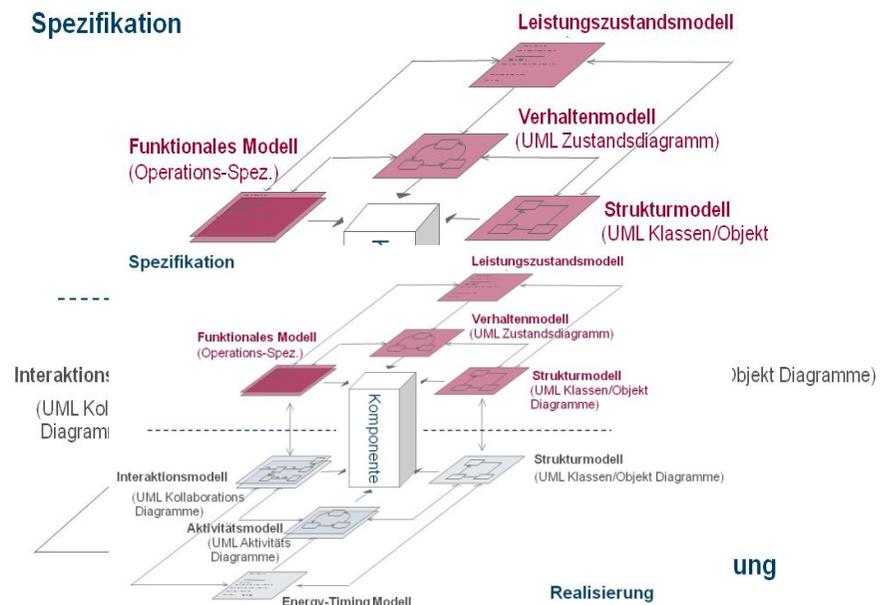
# Modellierung des Energiebedarf

- Modellierung des Energieverbrauchs mit Hilfe von Automaten
- Power State-Machines (Nieberg 2003)

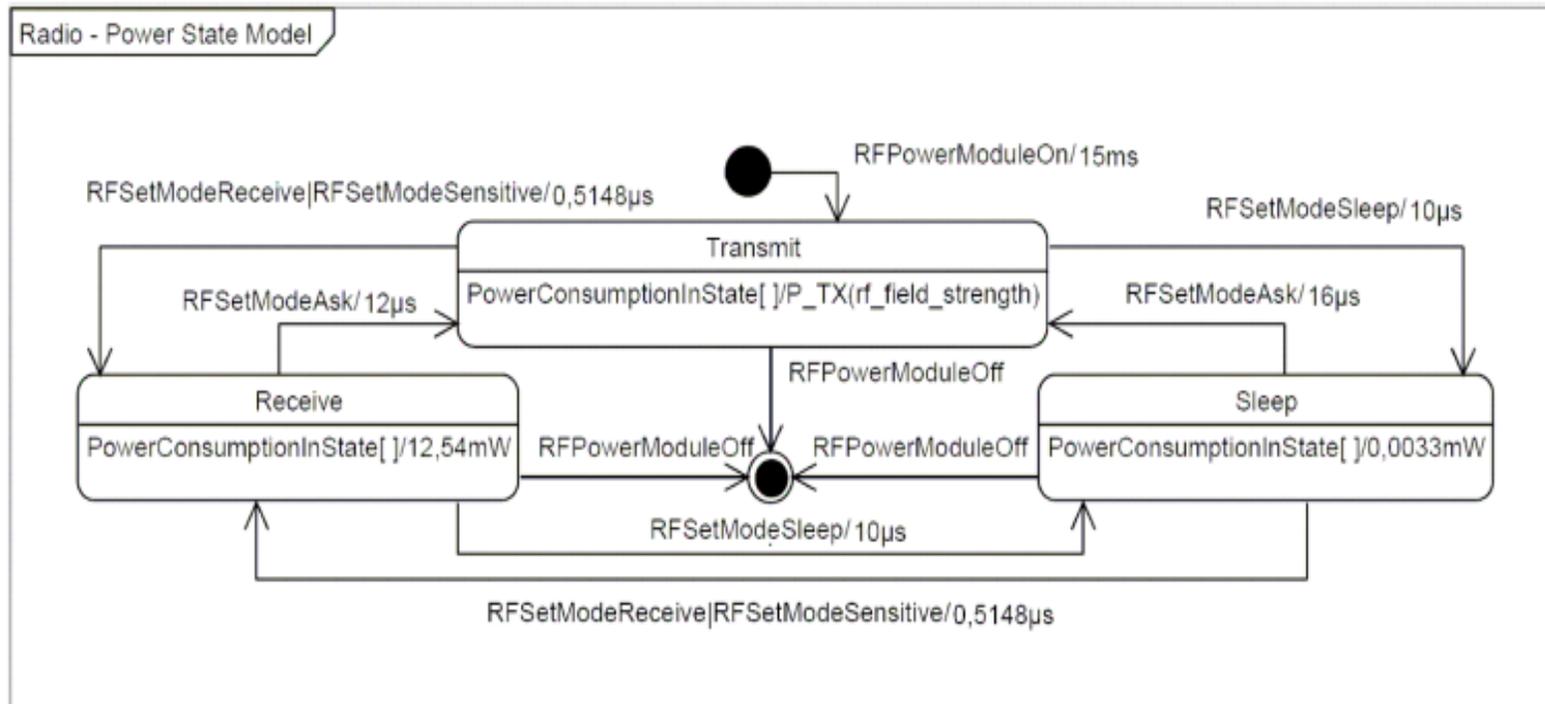


$$E = \sum_{j=1}^k t_j P_j + \sum_{\substack{i,j=1 \\ i \neq j}}^k s t_{ij} E_{ij}$$

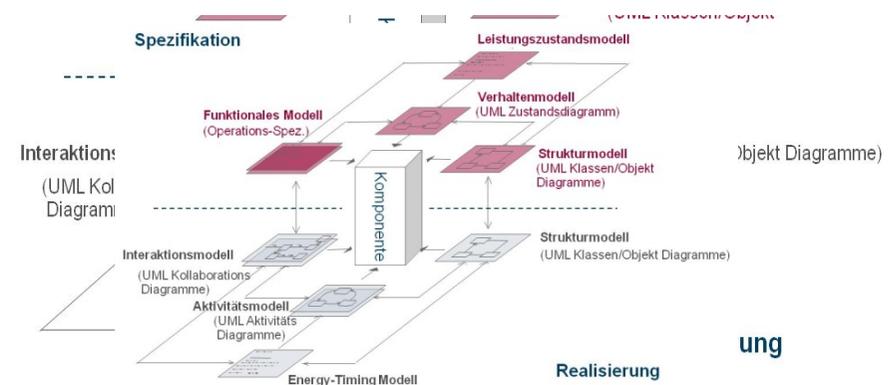
- Modellierung



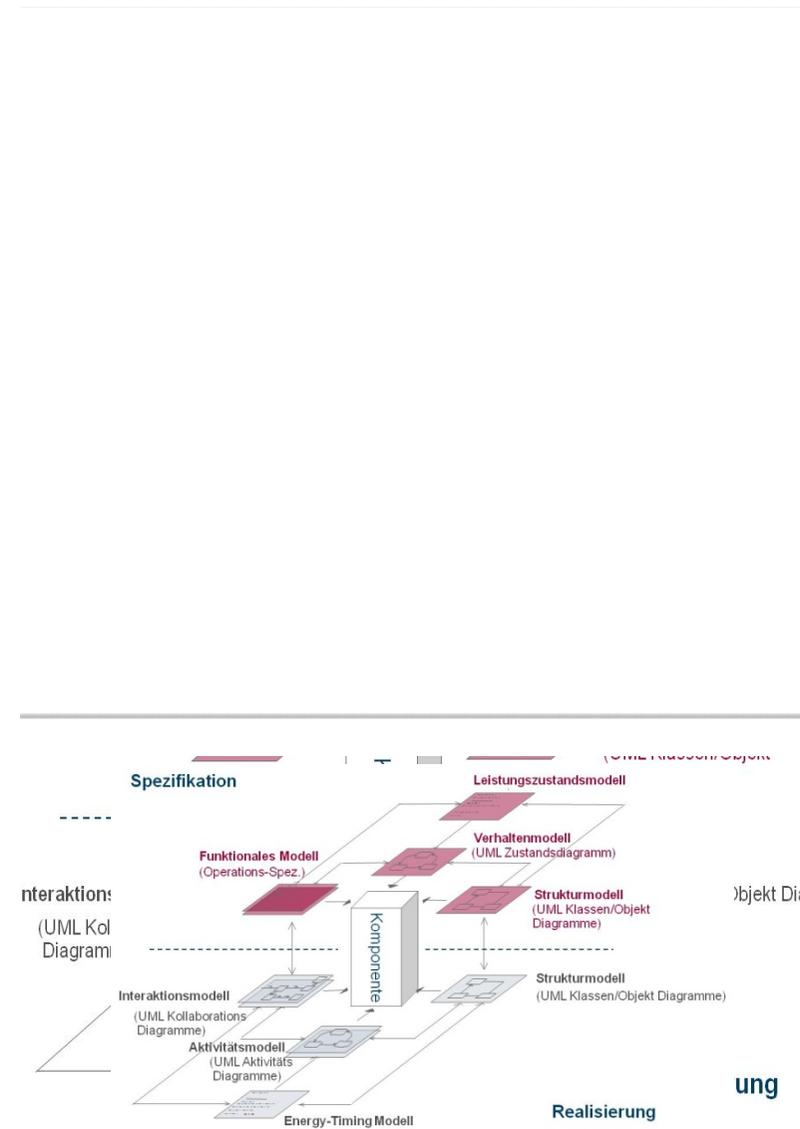
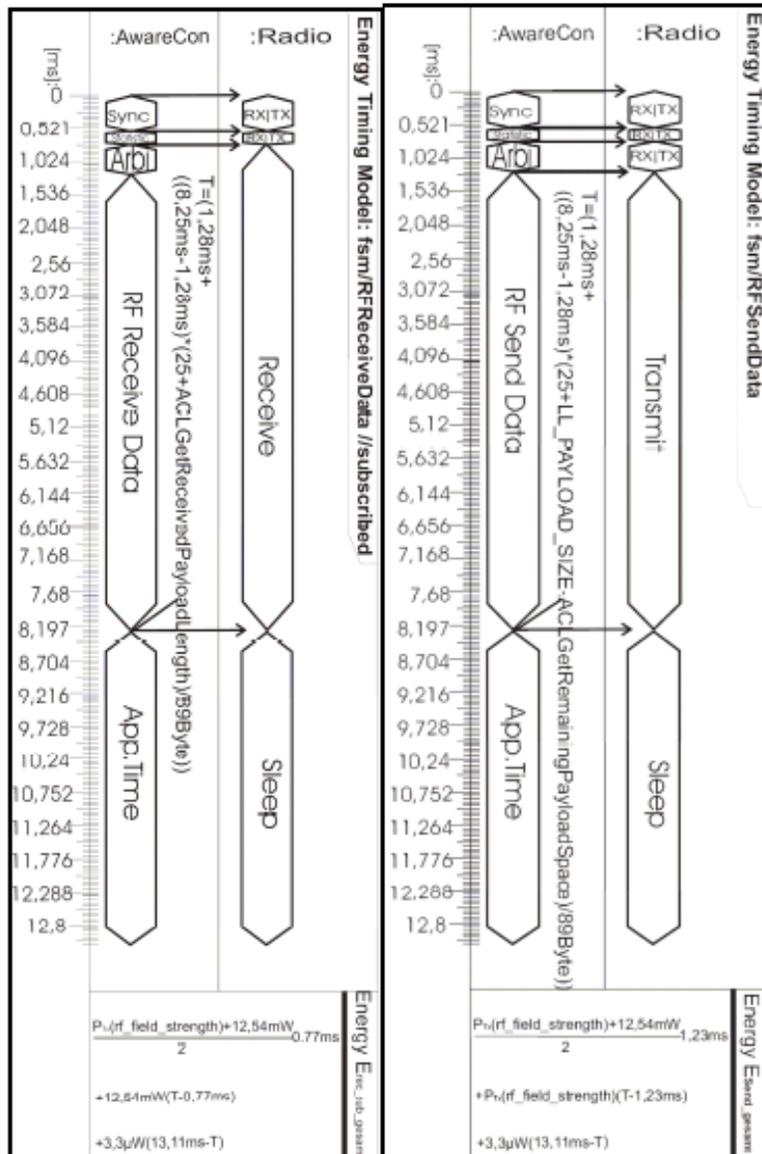
# Modellierung des Energiebedarf



- Modellierung



# Modellierung des Energiebedarf



# Optimieren und Vorhersagen

## ◆ Statisch

- Entwicklungszeit (Modelle, Code, ...)
- (Semi-)Manuelle Optimierung von Modellen und Code
  - ◆ Pattern-/Antipattern Technologie (basierend auf SORT)
- Vielversprechende Resultate [Domis06]
- Einschränkung auf vorhersagbare nicht komplexe Kommunikation

## ◆ Dynamisch

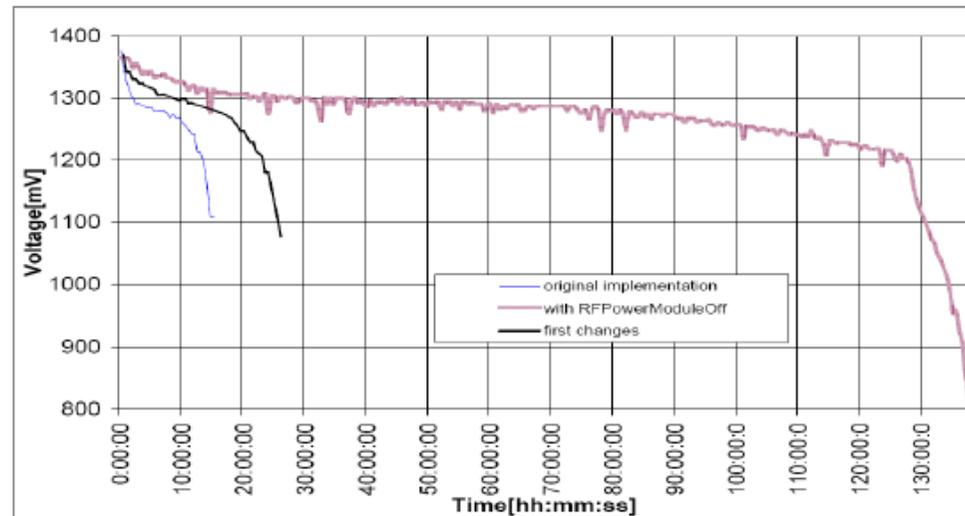
- Zur Laufzeit (Selbst-Adaption)
- Anwendbar für komplexe nicht vorhersagbare Kommunikation
- Einsatz einer Energie-Management Komponente
- Ergebnisse: Optimierung ähnlich/besser als statisches Verfahren

## ◆ Vorhersage des Energiebedarfs

- Verfolgte Idee: Nutzen des Palladio-Ansatzes [Reussner] zur Vorhersage von Software-Performanz

# Statische Optimierung - Ergebnisse

## ◆ Fallstudie – „Intelligent Mug“ [Domis06]

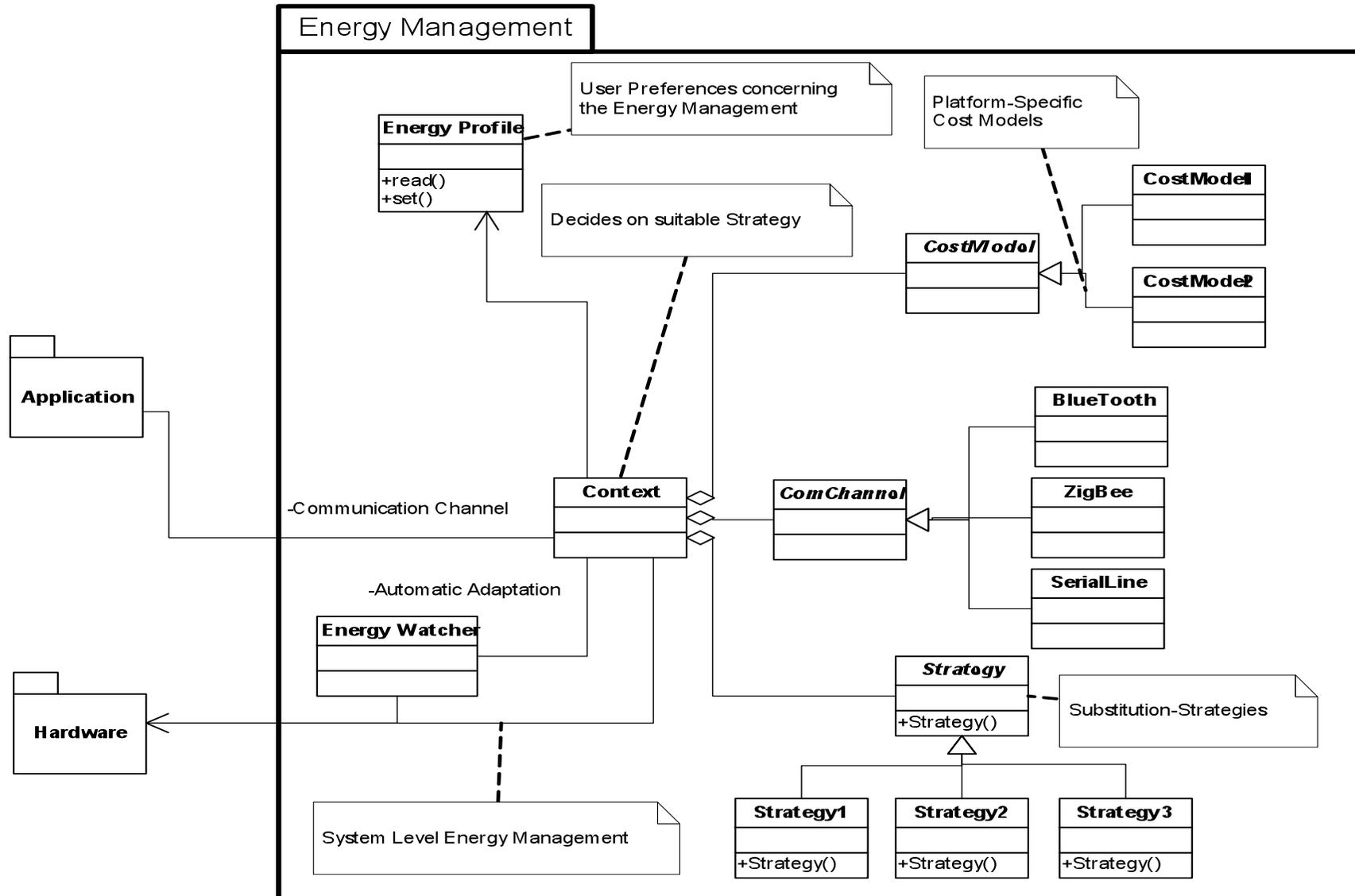


- ◆ Beschränkte Anzahl, Komplexität und Senderate von Nachrichten
- ◆ Hohe Anfangsinvestition
- Anwendbar für plattformspezifische- und kritische Systeme

# Dynamische Optimierung

- ◆ In mobilen Systemen sind Endgeräte, Services, etc. veränderbar
  - ◆ Statische Optimierung ist nicht immer möglich
  - ◆ Ansatz: Systeme müssen sich zur Laufzeit anpassen
- ◆ Adaptation zur Laufzeit erfordert Analyse und dynamischen Austausch
  - ◆ Keine vollständige technische Unterstützung
- ◆ Drahtlose Kommunikation ist der „größte“ energiebezogene Kostenfaktor (Stark 2002, Zhang 2005, Domis 2006, ...).
  - Entwicklung einer (Wrapper-) Komponenten als Manager
  - Anwendung von Ressourcen-Substitutionsstrategien
  - „Leicht“ integrierbar - MARMOT Komponente
- Annahmen
  - Wiederverwendung führt zu positiven ROI
  - Anfangsinvestitionen sind rasch amortisiert (Wiederverwendung und geringe Adaptionkosten)

# EMC – konzeptueller Aufbau



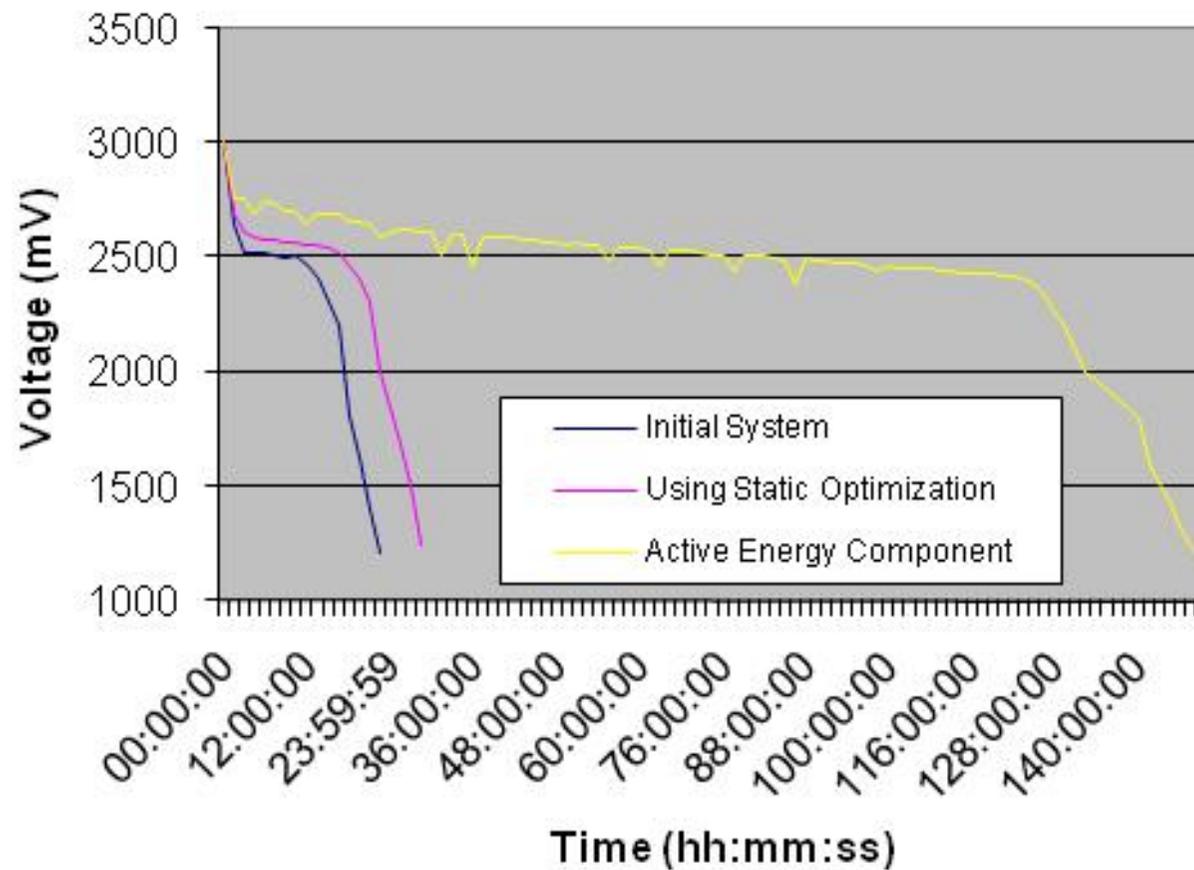
# Fallstudie – Wireless Sensor Network



- ◆ Controller, Router-, und End-Knoten
- ◆ Überwachung von Vital- und Geodaten bei Herdentieren
- ◆ Kommunikation mittels ZigBee und WLAN
- ◆ JN5139, 32-bit RISC, 16MIPS, 192kB ROM, 96kB RAM

# Messergebnisse – Batterie-Laufzeit

## Wireless Sensor Network - ZigBEE Communication



# Zusammenfassung

- ◆ Modellierung, Optimierung und Vorhersage des Energiebedarfs eingebetteter und mobiler Systeme
  - Nahtlose Integration in den Entwicklungsansatz
  - Ansätze zur statischen und dynamischen Optimierung
  - Offene Frage: verlässliche Vorhersage
  
- ◆ Ungelöste Probleme
  - Entwicklung einer standardisierten Messumgebung
  - Zeitaspekte (Spezifikation und Selbstadaptation)
  - Anwendung von Techniken der Aspektorientierung („Weaving“)
  - (Semi-) automatische Schwachstellenanalyse
  
- ◆ Aktuelle Arbeiten (Anwendung)
  - Optimierung von Sensorketten in Produktionsstraßen
  - Laufzeitoptimierung von „Handhelds“ im Straßenbau