

# Software, Performance and Resource Utilisation Metrics for Context-Aware Mobile Applications



von Christian Albrecht

# Inhalt

1. Einleitung
2. Spezifikation von Attributen und Metriken
3. Hypothesen + Empirische Evaluation
4. Modellierung
5. Praktische Anwendung
6. Zusammenfassung



# Einleitung

- Was bedeutet „Context-Aware“?
- mobile Anwendungen sind weit verbreitet → Qualität, besonders hinsichtlich der Effizienz wichtig
- Forschung und Entwicklung von Metriken bisher nur auf traditionelle Software beschränkt



# Spezifikation

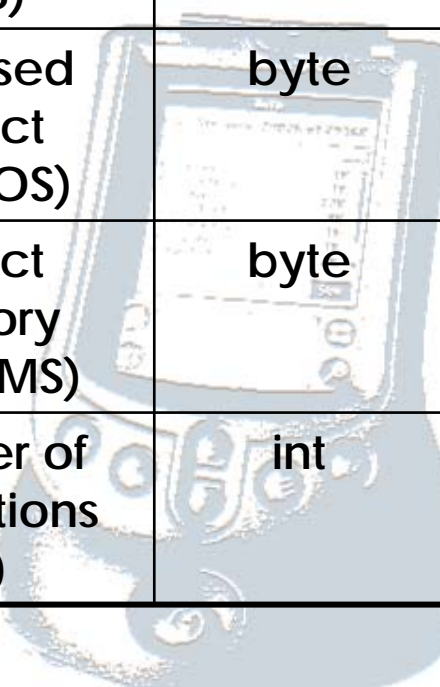
## 3 Arten von Attributen:

- Software
- Performance
- Resource Utilisation



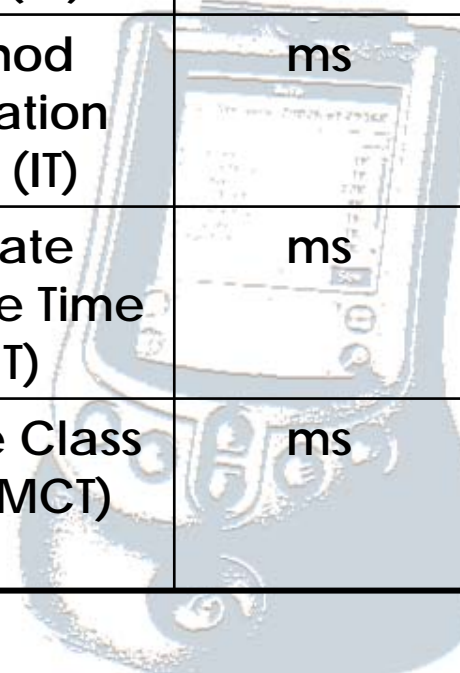
# Spezifikation | Software

Attribute	Definition	Metric	Unit
Object Compilation Volume	Size of an executable module	Executable Code Size (ECS)	byte
Object Serialisation Volume	Size of a serialised object	Serialised Object Size(SOS)	byte
Object Memory Volume	Size of an in-memory object	Object Memory Size(OMS)	byte
Method Invocation Frequenzy	Rate of occurrence of method invocation	Number of Invocations (NI)	int



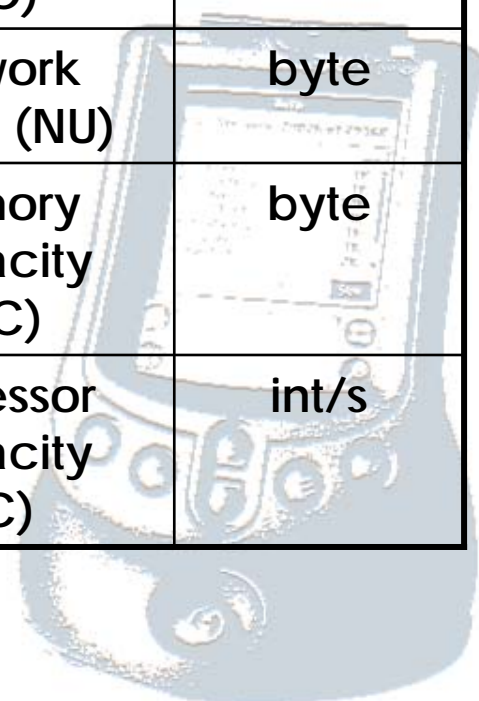
# Spezifikation | Performance-Attribute

Attribute	Definition	Metric	Unit
Method Execution Cost	Execution cost of a method	Method Execution Time (ET)	ms
Method Invocation Cost	The cost of calling a method	Method Invocation Time (IT)	ms
Object Migration Cost	The cost of moving an object instance between host	Migrate Instance Time (MIT)	ms
Class Migration Cost	The cost of moving an class implementation between host	Migrate Class Time (MCT)	ms



# Spezifikation | Resource Utilisation

Attribute	Definition	Metric	Unit
Network	The network bandwidth between hosts	Network Capacity (NC)	byte/s
Network Utilisation	The aggregate network bandwidth between two host	Network Usage (NU)	byte
Memory	The total memory available on a host	Memory Capacity (MC)	byte
Processor	The processing power of the host	Processor Capacity (PC)	int/s



# Hypothesen

16 Hypothesen zu den 3 Kategorien

Beispiel:

„Je größer das Object Compilation  
Volume, umso größer die Class  
Migration Cost“

[+ECS, +MCT]



# Empirische Evaluation

- via Experiment Daten gesammelt
- Test auf Client – Server – System
- zusätzlich auf anderer Hardware

Ergebnis:

- paarweise und kombinierte  
Abhängigkeiten der Metriken



# Modellierung

2 Strategien: „globale Anpassung“ versus  
„lokale Anpassung“

lokale Anpassung:

- Performance
- Ressource-Utilisation



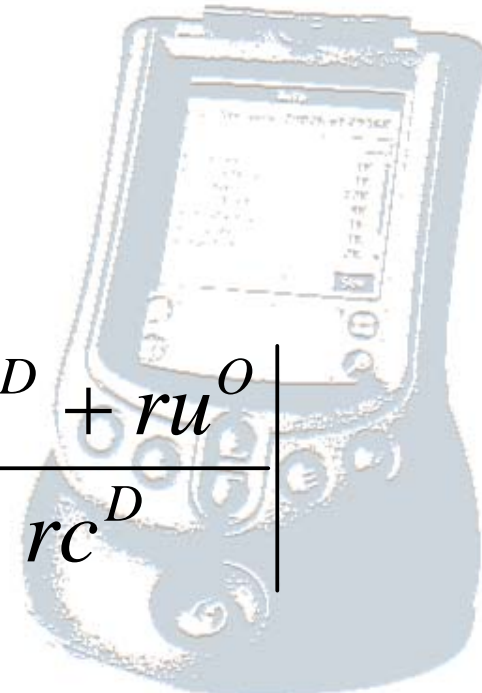
# Modellierung | lokale Anpassung

Performance:

$$rt_d = \sum_{i=1..m} \left[ NI_i * (rt_i^C - rt_i^D) \right] - mot$$

Ressource Utilisation:

$$ru_d = \left| \frac{ru^C}{rc^C} - \frac{ru^D}{rc^D} \right| - \left| \frac{ru^C - ru^O}{rc^C} - \frac{ru^D + ru^O}{rc^D} \right|$$



# Praktische Anwendung

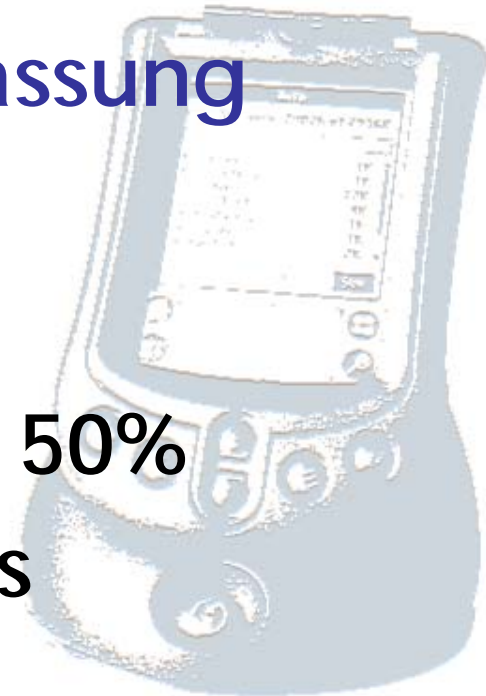
- Nutzung von MobJeX
- Anwendung: Taxi Dispatching System

Lokale und zufällige Anpassung

Ergebnis:

Resource Utilisation :25% versus 50%

Antwortzeit: 504ms versus 845ms



# Zusammenfassung

- Spezifikation von Metriken
- Hypothesen aufgestellt
- Empirisch evaluiert
- Mathematisches Modell
- In Framework implementiert
- Mehrwert festgestellt

Quelle:

C. Ryan und P. Rossi

,Software, Performance and Resource Utilisation Metrics for Context-Aware Mobile Applications, 2005

