

Pattern Based Re-Engineering: A Case Study

Integration-Engineering

Pascal Zipfel – AAM – SS 2009

Furtwangen, 28.05.2009

Agenda

- Die Autoren
- Hintergrund
- These
- Ausgangssystem
- Transformationsprozess
- Fazit

Die Autoren

- William C. Chu
- Chih-Wei Lu
- J.P. Shiu
- Xudong He

Hintergrund

- 1999 veröffentlicht
- Zitiert:
 - Marek Vokac, **Defect Frequency and Design Patterns: An Empirical Study of Industrial Code**, IEEE Transactions on Software Engineering, v.30 n.12, p.904-917, December 2004
 - M. Goedicke , U. Zdun, **Piecemeal legacy migrating with an architectural pattern language: a case study**, Journal of Software Maintenance: Research and Practice, v.14 n.1, p.1-30, January 2002
- Quellen
 - Design Patterns (Gamma, Helm, Johnson, Vlissides)
 - Weitere DP Bücher
 - Viel über Graphentheorie

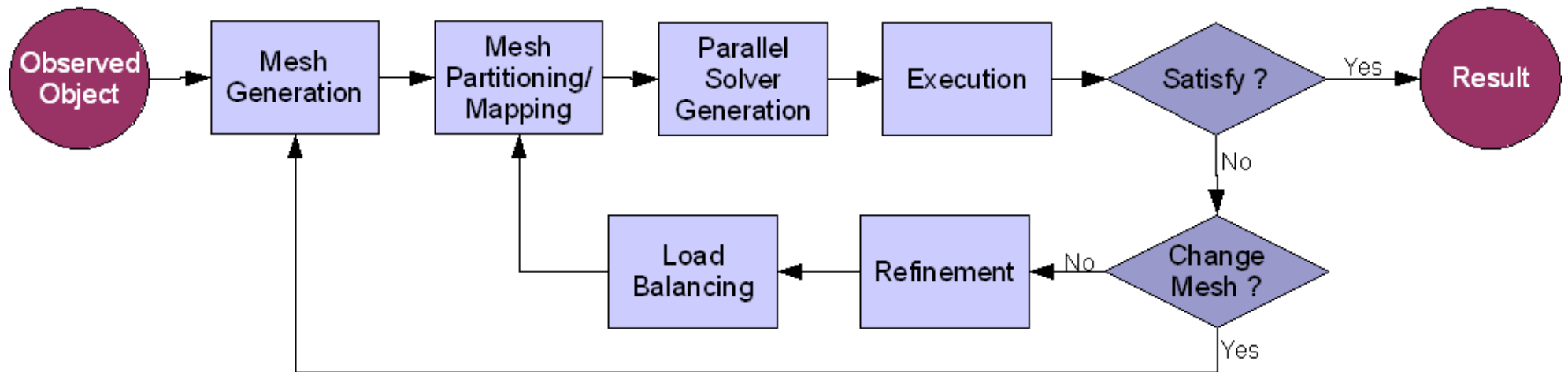
These

- Legacy-Software schwer wartbar
- Re-Engineering statt Erweiterung
- Standardisierung durch Einsatz von Design Pattern

Ausgangssystem

- Parallel Programm Generation Environment
- Unterstützung der Programmierer
- Optimieren von Endlichen Element Graphen
- Probleme:
 - Starke Fragmentierung
 - Algorithmen fest codiert
 - Protokolle fest codiert

Ausgangssystem



Re-Engineering Prozess

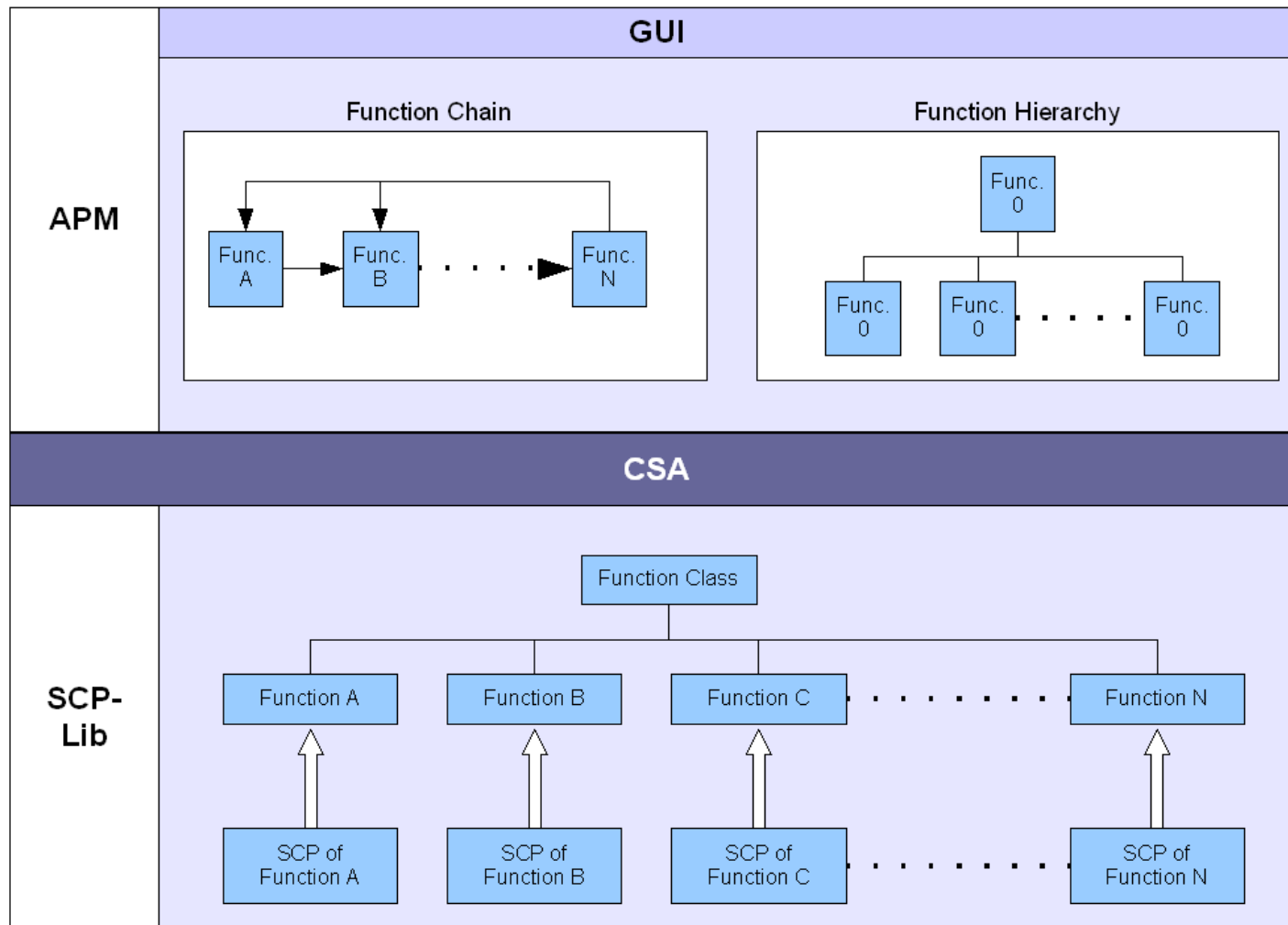
- 1) Analyse des Ausgangssystems
- 2) Definieren der Zielarchitektur
- 3) Klassifizieren der alten Strukturen (strukturell, funktionell oder erzeugend)
- 4) Zuordnen der alten Funktionen zu passenden Design Patterns
- 5) Re-Engineering des alten Systems mit Hilfe der Pattern

Analyse

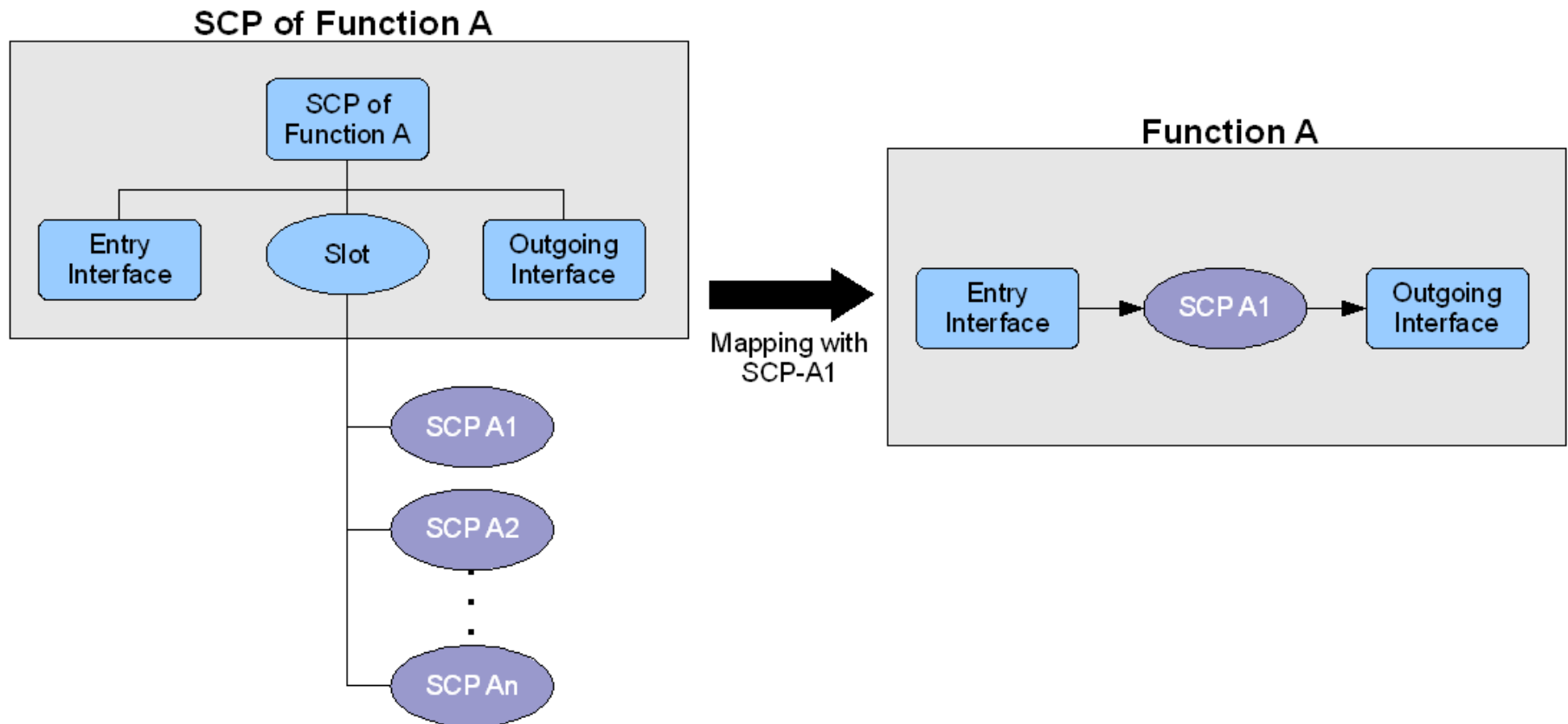
- Dokumentation
- Wissen der Entwickler/Designer
- Quellcode

- Aufteilen in Untersysteme
- Verhalten und Architektur herausfinden
- Zuteilung von Pattern

Zielarchitektur



Zielarchitektur



Design Pattern

- APM:
 - Strategy
 - Adapter
 - State
- CS-Agent
 - Builder
 - Factory

Erfahrungen

- Vorteile JAVA gegen C (Unix System)
 - Interoperabel
 - Konfigurierbar zur Laufzeit
 - OO Technologie gut geeignet für modulare SW
- Größte Schwierigkeit: Re-Coding
- Ein Untersystem nach dem anderen implementieren
- Leichter und schneller im weiteren Verlauf

Fazit

- Durchdachte, strukturierte Vorgehensweise
- Prinzipien heute noch anwendbar
- DP guter Weg, aber kein Allheilmittel
- Analyse oft sehr schwierig
- Oft zu teuer und zu aufwendig
- Ersetzen auf ein mal oft nicht möglich