Was ist Softwarearchitektur?

Ein Beispiel und drei Architekturen

Burkhardt Renz

Fachbereich MNI
Technische Hochschule Mittelhessen

Wintersemester 2017/18

„Architektur ist die Kombination von
utilitas, *firmitas* und *venustas*.“

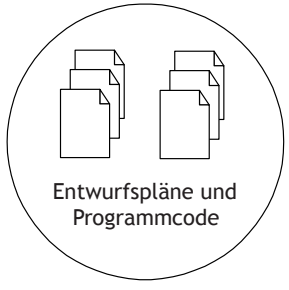
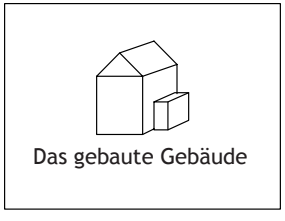
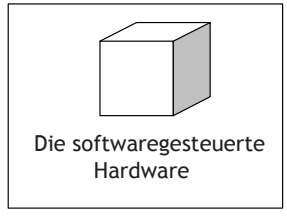
frei nach Vitruvius (Römischer Architekt 90-20 v. Chr.)

Softwarearchitektur will erreichen, dass das Anwendungssystem die Anforderungen erfüllt (utilitas), robust ist gegenüber Änderungen (firmitas) und eine gewisse Schönheit hat (venustas).

–

Fragt sich nur: wie?!

Eine (plakative) Gegenüberstellung

Gegenständliche Systeme	Softwarebestimmte Systeme
	
	
	

nach S. Wendt, HPI

Ergebnis der Gegenüberstellung

Gegenständliches System	**Softwaresystem**
• Grobstruktur mühelos sichtbar	• Grobstruktur nur mit besonderer Anstrengung sichtbar zu machen
• Feinstruktur nur mit großem Aufwand sichtbar zu machen	• Feinstruktur mühelos sichtbar

Was folgt daraus?

☞ Wir müssen die Struktur, den Aufbau von Software verständlich machen.
☞ Softwarearchitektur

Übersicht

- Ein Beispiel – drei Architekturen
  - Spezifikation
  - Objektorientierte Organisation
  - Pipes & Filters
  - Ereignisgesteuertes System
  - Architektur und Codestruktur
- Evolution und Architektur
- Was ist Softwarearchitektur?

Spezifikation von Keywords in Context (KWIC)

The KWIC (Key Word in Context) index system accepts an ordered set of lines; each line is an ordered set of words, and each word is an ordered set of characters. Any line may be "circulary shifted" by repeatedly removing the first word and appending it at the end of the line. The KWIC index system outputs a listing of all circular shifts of all lines in alphabetical order.

— *David Parnas*

Beispiel für KWIC

Input

```
Software Architecture in Practice
Bringing Design to Software
Problem Frames
```

Output

```
Architecture in Practice Software
Bringing Design to Software
Design to Software Bringing
Frames Problem
in Practice Software Architecture
Practice Software Architecture in
Problem Frames
Software Architecture in Practice
Software Bringing Design to
to Software Bringing Design
```

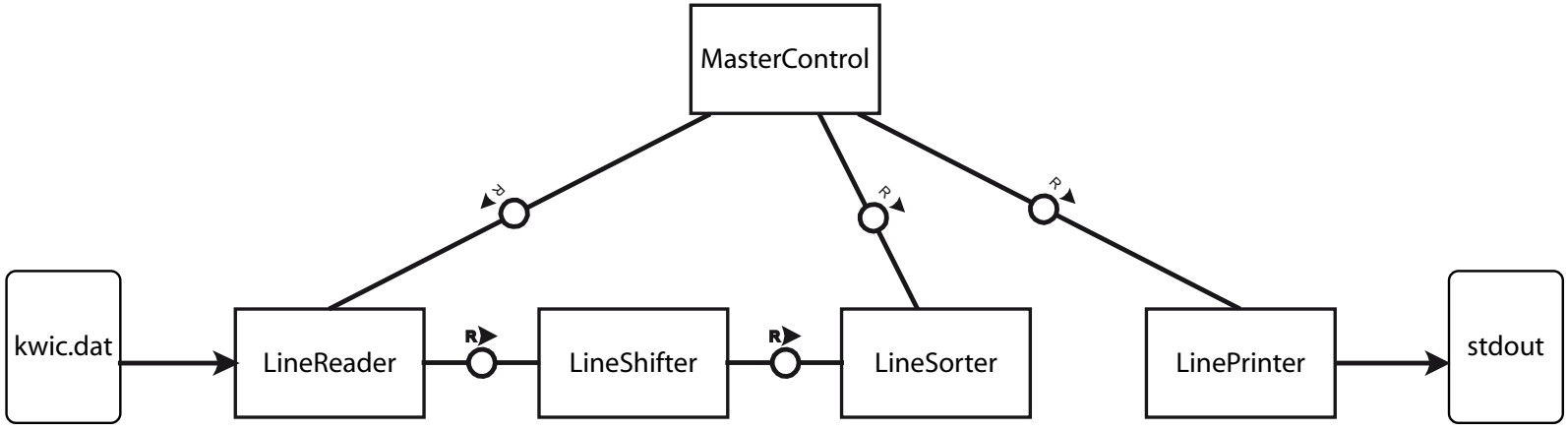
Diskussion

Wie kann man ein Programm strukturieren, das diese Spezifikation erfüllt?

Ad-hoc-Lösung

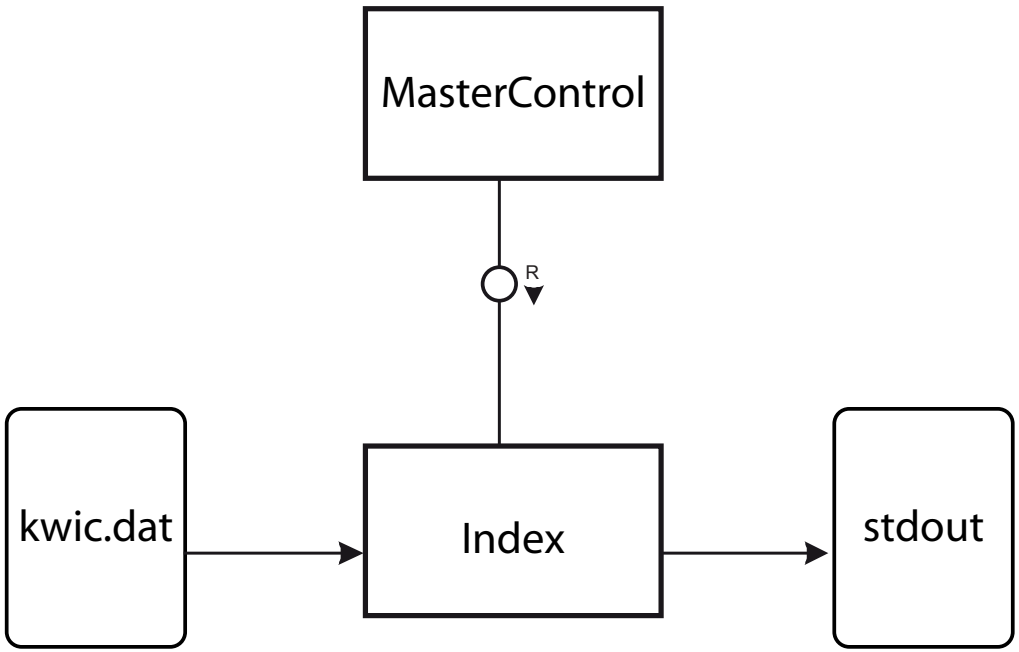
Demo

Struktur der Ad-hoc-Lösung

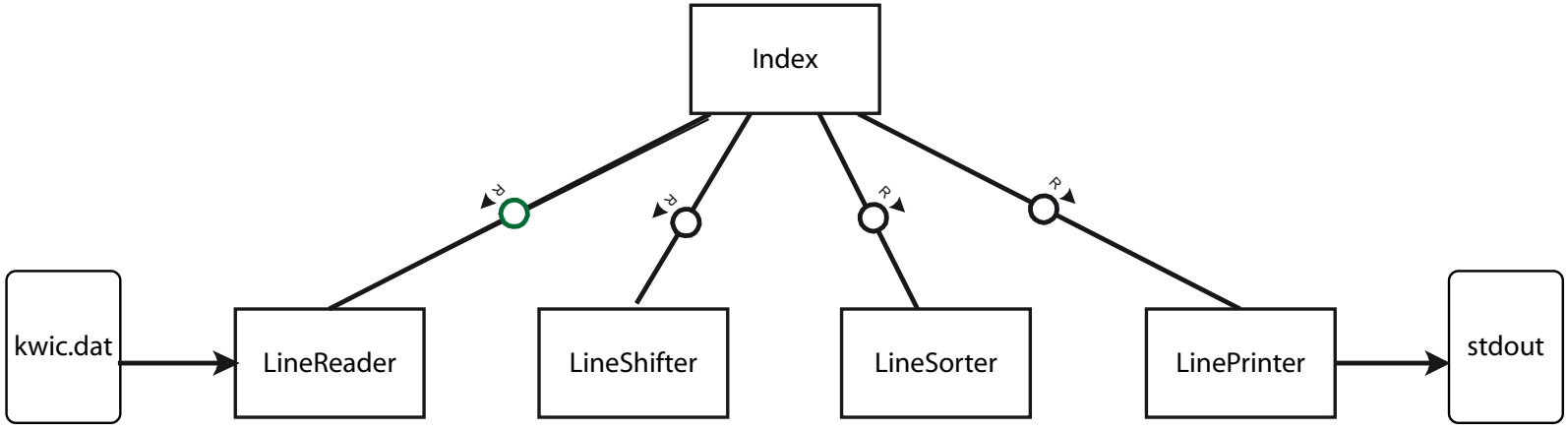


Objektorientierte Organisation

- Kapselung von Daten
- Methoden zur Manipulation der Daten



Struktur des Objektnetzes



Objektorientierte Organisation

Demo

Charakteristik

- Kontrollfluss durch zentrale Koordination
- Prozedurale Abstraktion durch Kapselung der Zugriffsmethoden
- Datenkapselung
- Algorithmen und Datenrepräsentation der einzelnen Klassen können geändert werden, ohne die anderen zu beeinflussen.
- „Dependency Injection“

Pipes & Filters

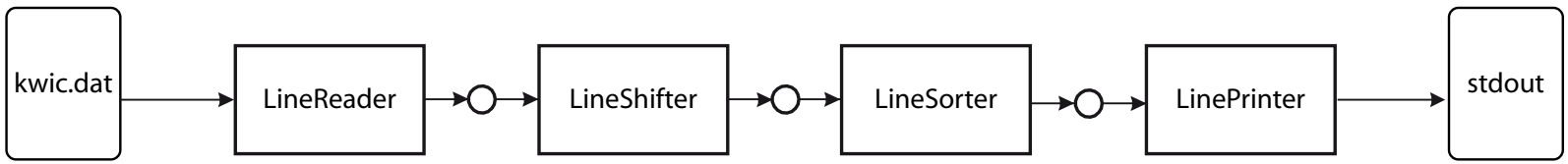
Demo 1 (Prozesse)

Demo 2 (Threads)

Demo 3 (Funktionaler Stil in Java 8)

Demo 4 (Reaktiver Stil in Java 8 mit RxJava)

Struktur


kwic.dat
LineReader
LineShifter
LineSorter
LinePrinter
stdout

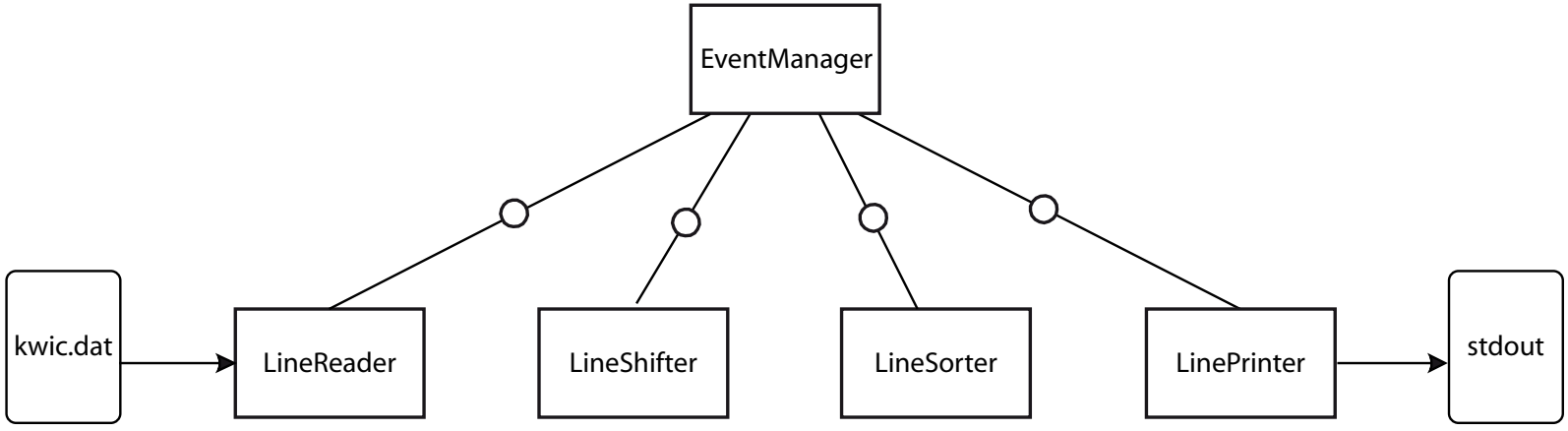
Charakteristik

- Die Steuerung ist verteilt: jeder Filter arbeitet, sobald Daten an seiner Eingabe anliegen.
- Die Filter sind voneinander unabhängig, sie kommunizieren nur über die Pipes zwischen ihnen.
- Datenfluss und Kontrollfluss fallen zusammen

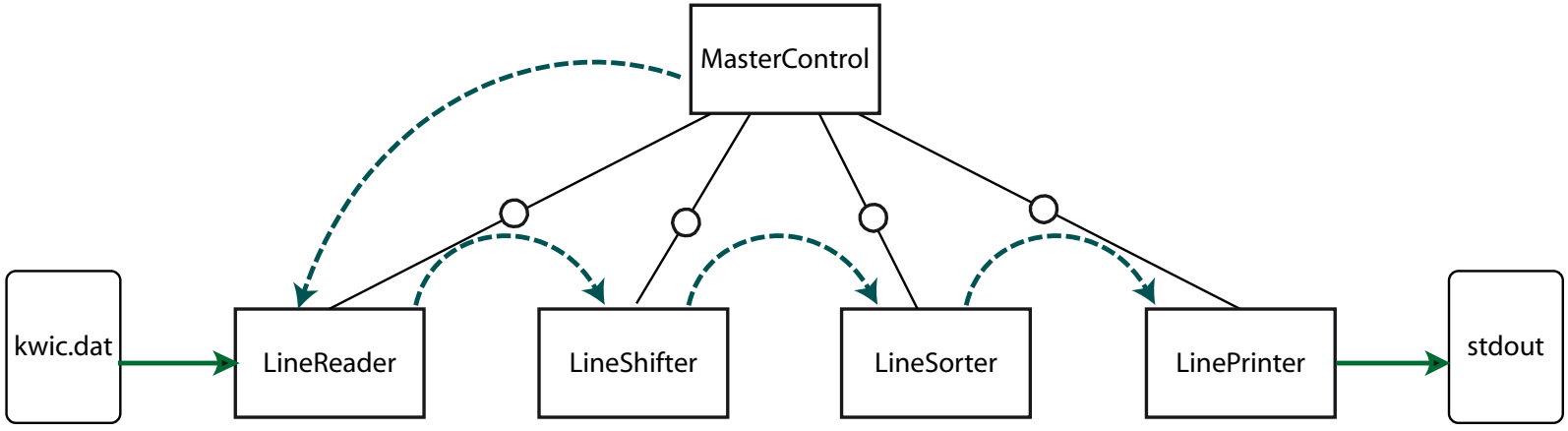
Ereignisgesteuertes System mit indirektem Aufruf

Demo

Struktur


EventManager
kwic.dat
LineReader
LineShifter
LineSorter
LinePrinter
stdout

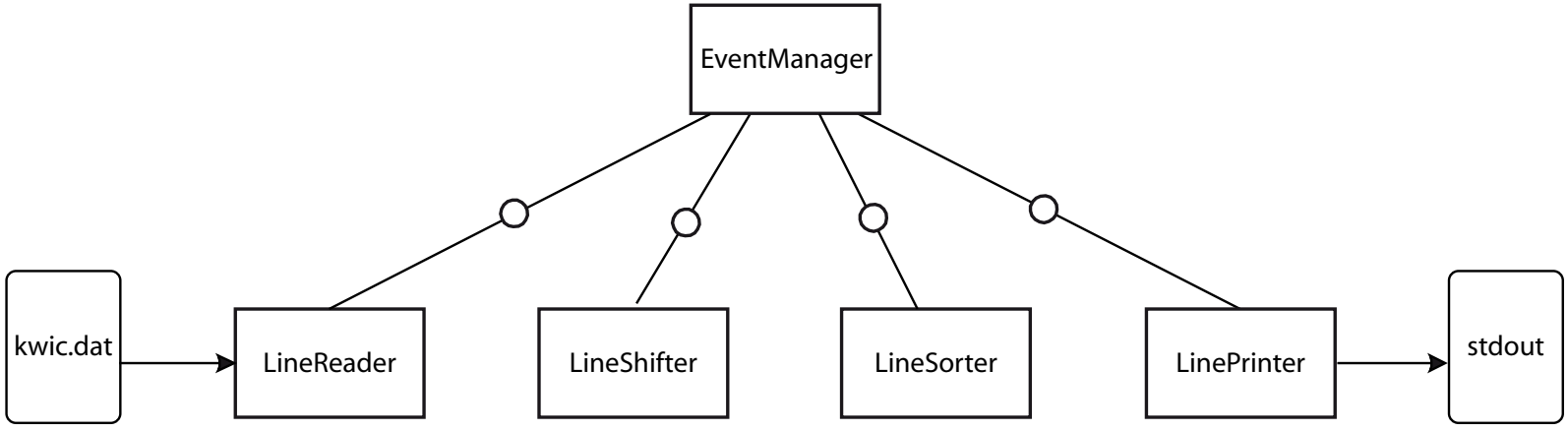
Laufzeitstruktur



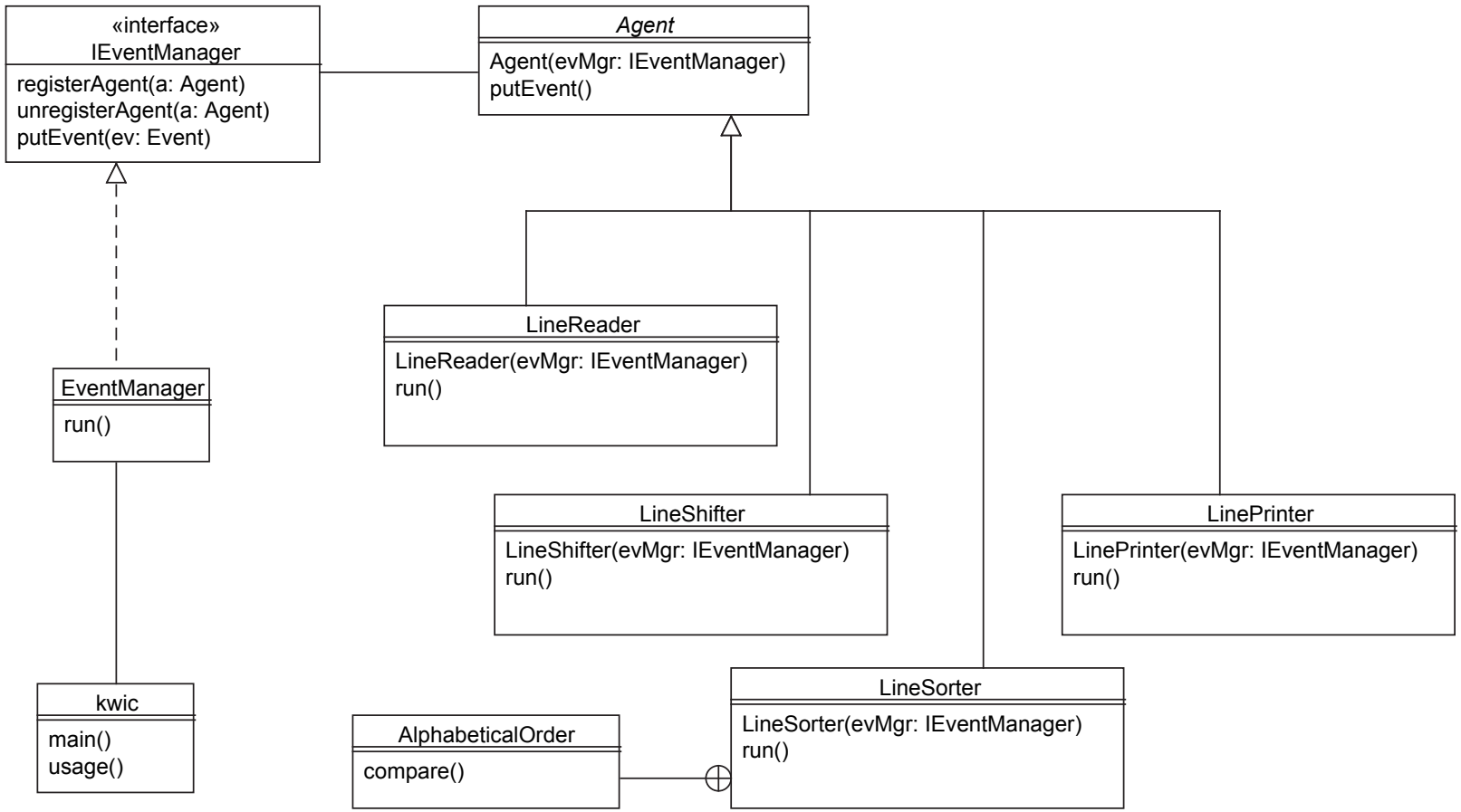
Charakteristik

- Keine zentrale Steuerung.
- Events führen zu Aktionen, die ihrerseits Events erzeugen.
- Veränderung der Daten löst Events aus.
- Modell „aktiver Daten“

Aufbaustruktur von KWIC in Variante 3



Codestruktur von KWIC in Variante 3



Übersicht

- Ein Beispiel – drei Architekturen
- Evolution und Architektur
  - Neue Anforderungen
  - Vergleich der Lösungen
  - Fazit
- Was ist Softwarearchitektur?

Neue Anforderungen an KWIC

1. Nicht-Berücksichtigung bestimmter Wörter (wie „der, die, das“) für den Index.
2. Input nicht nur aus einer, sondern aus mehreren Quellen.
3. Graphische Benutzeroberfläche für die Abfrage des Index.

Neue Anforderungen an KWIC

Diskussion

Objektorientierte Organisation

1. Implementierung einer neuen LineShifter-Implementierung, die dann dem Index-Objekt injiziert wird.
2. Paralleler Input nur möglich, indem die Struktur grundlegend geändert wird.
3. Die Implementierung von Index zum Schreiben eines Index verwenden, GUI objektorientiert entwickeln, IndexReader schreiben und dort verwenden.

Pipes & Filters

1. Filter für „StopWords“ zwischenschalten.
2. Paralleler Input leicht möglich, vorausgesetzt man verwendet Pipes, die mehrere Eingaben mischen können.
3. Die Implementierung von KWIC mit Pipes & Filters zum Schreiben eines Index verwenden, GUI objektorientiert entwickeln, IndexReader schreiben und dort verwenden.

Ereignisgesteuertes System

1. Filter für „StopWords“ in das Protokoll einbeziehen – Änderung an anderen Filtern notwendig.
   Oder neuen LineShifter schreiben – ohne das Protokoll zu verändern.
2. Paralleler Input leicht möglich: die parallelen Reader verwenden alle dasselbe Protokoll. Aber: Ende der Verarbeitung erst, wenn sich der letzte Reader abgemeldet hat.
3. Die Implementierung von Index durch das ereignisgesteuerte System zum Schreiben eines Index verwenden, GUI objektorientiert entwickeln, IndexReader schreiben und dort verwenden.

Fazit

1. Architektur $\perp$ Funktionalität
2. Architektur $\Rightarrow$ Qualität
3. Architektur $\neq$ Codestruktur

Übersicht

- Ein Beispiel – drei Architekturen
- Evolution und Architektur
- Was ist Softwarearchitektur?
  - Definition
  - Wozu Softwarearchitektur?
  - Literatur & Links

Was ist Softwarearchitektur? – Definition

The software architecture of a system is the set of structures needed to reason about the system, which comprise software elements, relations among them, and properties of both.

— *Len Bass et al.*

Elemente der Definition

Elemente der Softwarearchitektur

- Elemente, *Komponenten* – definieren den Ort von Berechnungen oder auch Speicher, z.B. Filter, Datenbanken, Objekte, Server, Klienten usw.
- Beziehungen, *Konnektoren* – definieren die Interaktion der Komponenten, z.B. Funktionsaufruf, Pipe, Event usw.
- Eigenschaften – definieren Vorgaben und Beschränkungen für Komponenten, z.B. Schnittstellen, Qualitätsmerkmale usw.

Diskussion

- Abstraktion: *Wesentliche* Elemente und ihre Beziehungen
- Struktur: Zerlegung und Zusammenbau
- Struktur*en*: Verschiedene Sichten auf das Wesentliche des Systems
- Beziehungen: Organisierende Prinzipien des Zusammenwirken der Elemente
- Plan: *Bauplan* im Unterschied zu Dokumentation und Projektplan

Was ist Softwarearchitektur? – Kritik der Definition

The architecture of a software system defines that system in terms of components and of interactions among those components. In addition to specifying the structure and topology of the system, the architecture shows the intended correspondence between the system requirements and elements of the constructed system.

— *Mary Shaw et al.*

What is surprising about the Shaw et al. model [in diesem Geiste ist auch die Definition von Len Bass] is that, rather than defining the software's architecture as existing within the software, it is defining a description of the software's architecture as if that were the architecture

— *Roy Thomas Fielding*

Welche Strukturen?

Software ist die Beschreibung des gewollten Verhaltens einer universellen, abstrakten Maschine (Computer).

Beschreibung – Beschriebenes – Umgebung

1. Struktur der Beschreibung: Codestruktur
2. Struktur des Beschriebenen: konzeptionelle Struktur des (laufenden) Systems
3. Struktur der Umgebung: Infrastruktur

Architektonische Themen – Auswahl

- Regeln für „Querschnittsthemen“
  z.B. Fehlerbehandlung, Verteilung...
- Richtlinien für Bauplan
- Fachliche Architektur
  z.B. Strukturen des Anwendungsgebiets
- Qualitätsanforderungen und Strategien, sie zu erreichen
  z.B. Erweiterbarkeit, Anpassbarkeit des Systems
- Infrastruktur, eingesetzte Technik
  z.B. Frameworks, Sprachen, Datenbanken...

Die Abgrenzung der architekturell relevanten Strukturen ist die zentrale Aufgabe des Architekten und kann nicht nach formalen Kriterien vollzogen werden

— *Peter Tabeling*

Aufgaben und Ziele

- Mittel der Kommunikation und Diskussion aller Interessengruppen (*Stakeholder*)
- Festlegung der ersten und grundlegenden Entscheidungen für Design und Implementierung (*Constraints*)
- Grundlage für das Erreichen der funktionalen und qualitativen Merkmale des zu konstruierenden Systems
- Architektur fungiert als „mentaler Prototyp“
- Architektur(-stile, -muster) sind wiederverwendbar, auf andere Systeme übertragbar

Erwartungen an eine Softwarearchitektur

- Klare Entwurfsabsichten, damit die intendierte Architektur erhalten und verständlich bleibt
- Designzentren festlegen
- Basis für Design, damit die wesentlichen Fragen gestellt werden
- Verbesserung der Änderbarkeit, indem die Designzentren erkennbar bleiben und erhalten werden

Grundlegende Entscheidungen in der Architektur

Funktionalität Fähigkeit des Systems, die intendierte Aufgabe zu erfüllen (kann durch verschiedene Architekturen erreicht werden)

Qualitätsmerkmale Insbesondere nicht-funktionale Merkmale, die in hohem Maße von der Architektur beeinflusst werden

Organisation Architektur bestimmt auch die Organisation des Entwicklungsprozesses

Eckpfeiler Architektur legt den Rahmen fest, in dem sich Design und Implementierung bewegen (z.B. Framework)

Zusammenfassung: Aufgaben von Softwarearchitektur

1. Verstehbare Darstellung der zentralen Entwurfsentscheidungen
2. Wiederverwendbarkeit von Komponenten eines Systems
3. Grundlegender Bauplan für die Konstruktion eines Softwaresystems
4. Basis für kontrolllierte Evolution eines Systems oder einer Produktlinie
5. Grundlage für die Analyse eines (evtl noch gar nicht gebauten) Softwaresystems – z.B. in Bezug auf Qualitätsmerkmale
6. Mittel des Managements der Entwicklung und Weiterentwicklung

Fazit: Was ist Softwarearchitektur?

To be architectural is to be the most abstract depiction of the system that enables reasoning about critical requirements and constrains all subsequent refinements.

— *Mark Klein*

A mental prototype is a model of the system which outlines a potential system solution. It describes functionality, but leaves open most of the implementation details.

— *Andreas Knöpfel, Bernhard Gröne, Peter Tabeling*

Literatur


Mary Shaw, David Garlan
Software Architecture: Perspectives on an Emerging Discipline
Upper Saddle River, NJ: Prentice-Hall, 1996.


Len Bass, Paul Clements, Rick Kazman
Software Architecture in Practice
Boston: Addison-Wesley, Third Edition 2012.


Jan Bosch
Design and Use of Software Architectures
Harlow, UK: Addison-Wesley, 2000.


Christine Hofmeister, Robert Nord, Dili Soni
Applied Software Architecture
Reading, MA: Addison-Wesley, 2000.

Literatur


Ian Gorton
Essential Software Architecture
Berlin: Springer, Second Edition 2011.


Peter Tabeling
Softwaresysteme und ihre Modellierung
Berlin: Springer, 2006.


Richard N. Taylor, Nenad Medvidović, Eric M. Dashofy
Software Architecture: Foundations, Theory, and Practice
John Wiley & Sons, 2010.


George Fairbanks
Just Enough Software Architecture: A Risk-Driven Approach
Boulder CO: Marshall & Brainerd, 2010.

Links

- Software Engineering Institute
  Software Architecture
  http://www.sei.cmu.edu/architecture/
- Michael Stal, Stefan Tilkov, Markus Völter, Christian Weyer
  SoftwareArchitekTOUR - Podcast für den professionellen Softwarearchitekten
  http://www.heise.de/developer/podcast/