

# L<sup>A</sup>T<sub>E</sub>X, beamer, tikz und Co.

## 17. Das beamer Paket – dynamische Aspekte

Thomas Worsch

Fakultät für Informatik  
Karlsruher Institut für Technologie

Wintersemester 2016/2017

Lokale dynamische Struktur: Overlays

Globale dynamische Struktur: „Hin- und Her-Springen“

Hypertargets und Hyperlinks

„Hineinzoomen“ in Bilder

## Strukturen

- ▶ globale statische logische Struktur
  - ▶ Aufbau der Quelldatei
- ▶ globale dynamische Struktur
  - ▶ teils durch die statische Struktur festgelegt
  - ▶ teils durch die Reihenfolge der „Knopfdrücke“
- ▶ lokale statische logische Struktur
- ▶ lokale dynamische Struktur
- ▶ Stile
  - ▶ für die diversen Strukturelemente
  - ▶ für die diversen Layoutelemente

# Überblick

Lokale dynamische Struktur: Overlays

Globale dynamische Struktur: „Hin- und Her-Springen“

## Sukzessives „Aufdecken“ von Teilen eines Frames

- ▶ unbedingt mit großer Zurückhaltung verwenden
- ▶ denn
  - ▶ es nervt den Vortragenden (viel Knöpfedrücken)
  - ▶ es nervt schnell die Zuhörer (Bevormundung?)
- ▶ es zwingt den Vortragenden, sich beim Reden an die Reihenfolge auf der Folie zu halten

## Sukzessives „Aufdecken“ von Teilen eines Frames

- ▶ **unbedingt** mit großer Zurückhaltung verwenden
- ▶ denn
  - ▶ es nervt den Vortragenden (viel Knöpfedrücken)
  - ▶ es nervt schnell die Zuhörer (Bevormundung?)
  - ▶ es zwingt den Vortragenden, sich beim Reden an die Reihenfolge auf der Folie zu halten

## Sukzessives „Aufdecken“ von Teilen eines Frames

- ▶ **unbedingt mit großer Zurückhaltung verwenden**
- ▶ denn
  - ▶ es nervt den Vortragenden (viel Knöpfedrücken)
  - ▶ es nervt schnell die Zuhörer (Bevormundung?)
  - ▶ es zwingt den Vortragenden, sich beim Reden an die Reihenfolge auf der Folie zu halten

## Sukzessives „Aufdecken“ von Teilen eines Frames

- ▶ unbedingt mit großer Zurückhaltung verwenden
- ▶ denn
  - ▶ es nervt den Vortragenden (viel Knöpfedrücken)
  - ▶ es nervt schnell die Zuhörer (Bevormundung?)
  - ▶ es zwingt den Vortragenden, sich beim Reden an die Reihenfolge auf der Folie zu halten



## Sukzessives „Aufdecken“ von Teilen eines Frames

- ▶ unbedingt mit großer Zurückhaltung verwenden
- ▶ denn
  - ▶ es nervt den Vortragenden (viel Knöpfedrücken)
  - ▶ es nervt schnell die Zuhörer (Bevormundung?)
  - ▶ es zwingt den Vortragenden, sich beim Reden an die Reihenfolge auf der Folie zu halten

## Sukzessives „Aufdecken“ von Teilen eines Frames

- ▶ unbedingt mit großer Zurückhaltung verwenden
- ▶ denn
  - ▶ es nervt den Vortragenden (viel Knöpfedrücken)
  - ▶ es nervt schnell die Zuhörer (Bevormundung?)
  - ▶ es zwingt den Vortragenden, sich beim Reden an die Reihenfolge auf der Folie zu halten

## Sukzessives „Aufdecken“ von Teilen eines Frames

- ▶ unbedingt mit großer Zurückhaltung verwenden
- ▶ denn
  - ▶ es nervt den Vortragenden (viel Knöpfedrücken)
  - ▶ es nervt schnell die Zuhörer (Bevormundung?)
- ▶ es zwingt den Vortragenden, sich beim Reden an die Reihenfolge auf der Folie zu halten

## Das `\pause` Kommando

ein oder mehrere Vorkommen von `\pause` in einem `frame`

- ▶ bewirken die Erzeugung mehrerer *slides*
- ▶ auf Slide 1 sieht man (im wesentlichen) nur alles vor dem 1. `\pause`
- ▶ auf Slide 2 sieht man (im wesentlichen) nur alles vor dem 2. `\pause`
- ▶ usw.
- ▶ auf Slide  $n + 1$  sieht man alles

zum Beispiel liefert:

```
slide 1 \pause slide 2 \pause slide 3 \pause slide 4  
slide 1 slide 2 slide 3 slide 4
```

## Das `\pause` Kommando

ein oder mehrere Vorkommen von `\pause` in einem `frame`

- ▶ bewirken die Erzeugung mehrerer *slides*
- ▶ auf Slide 1 sieht man (im wesentlichen) nur alles vor dem 1. `\pause`
- ▶ auf Slide 2 sieht man (im wesentlichen) nur alles vor dem 2. `\pause`
- ▶ usw.
- ▶ auf Slide  $n + 1$  sieht man alles

zum Beispiel liefert:

```
slide 1 \pause slide 2 \pause slide 3 \pause slide 4  
slide 1 slide 2 slide 3 slide 4
```

## Das `\pause` Kommando

ein oder mehrere Vorkommen von `\pause` in einem `frame`

- ▶ bewirken die Erzeugung mehrerer *slides*
- ▶ auf Slide 1 sieht man (im wesentlichen) nur alles vor dem 1. `\pause`
- ▶ auf Slide 2 sieht man (im wesentlichen) nur alles vor dem 2. `\pause`
- ▶ usw.
- ▶ auf Slide  $n + 1$  sieht man alles

zum Beispiel liefert:

```
slide 1 \pause slide 2 \pause slide 3 \pause slide 4  
slide 1 slide 2 slide 3 slide 4
```

## Das `\pause` Kommando

ein oder mehrere Vorkommen von `\pause` in einem `frame`

- ▶ bewirken die Erzeugung mehrerer *slides*
- ▶ auf Slide 1 sieht man (im wesentlichen) nur alles vor dem 1. `\pause`
- ▶ auf Slide 2 sieht man (im wesentlichen) nur alles vor dem 2. `\pause`
- ▶ usw.
- ▶ auf Slide  $n + 1$  sieht man alles

zum Beispiel liefert:

```
slide 1 \pause slide 2 \pause slide 3 \pause slide 4  
slide 1 slide 2 slide 3 slide 4
```

## Das `\pause` Kommando (2)

- ▶ intern wird der Zähler `beamerpauses` verwendet
- ▶ initialer Wert: 1
- ▶ durch jedes Kommando `\pause` wird er um 1 erhöht
  
- ▶ für jeden vorkommenden Zählerwert wird eine Dokumentenseite erzeugt
- ▶ bei der Erzeugung der  $i$ -ten Seite wird nur das berücksichtigt, was vor dem  $i$ -ten `\pause`-Kommando steht



## Das `\pause` Kommando (3)

- ▶ allgemeine Syntax: `\pause` [*⟨Nummer⟩*]
- ▶ setzt den Zähler `beamerpauses` auf *⟨Nummer⟩*

### Implementierung (näherungsweise)

- ▶ `beamerpauses` auf neuen Wert *x* setzen
  - ▶ `\pause` ersetzen durch `\onslide<x->`
- 
- ▶ *<x->* ist eine sogenannte *⟨Overlay Spezifikation⟩*
  - ▶ `\onslide` kommt gleich noch

## Overlay Spezifikationen

- ▶ in spitzen Klammern
- ▶ einfacher Fall: Liste von
  - ▶ Nummern einzelner Slides
  - ▶ Intervalle von Slidenummern
- ▶ Beispiel `<-3,5-7,9,13->`
  
- ▶ so etwas ist nicht nur bei `\uncover` und Co. möglich
- ▶ `beamer` definiert diverse Kommandos und Umgebungen um, so dass sie auf Overlay Spezifikationen reagieren, z. B.
  - ▶ `\item`
  - ▶ `\textcolor`, `\textbf`, ...  
Kommando nur „aktiv“, wenn aktuelle Slidenummer passt

`\uncover`, `\visible`, `\only`, `\alt`, `\temporal`

current slide 1

sl.1: `\uncover<2-3>{uncovered on slides 2-3}`

\_\_\_\_\_uncovered on slides 2-3\_\_\_\_\_

sl.1: `\visible<2-3>{visible on slides 2-3}`

\_\_\_\_\_

sl.1: `\only<2-3>{only on 2-3}`

\_\_\_\_\_

sl.1: \_\_\_\_\_ `\alt<2-3>{on 2-3}{not on 2-3}` \_\_\_\_\_

\_\_\_\_\_not on 2-3\_\_\_\_\_

sl.1: `\temporal<2-3>{before 2}{on 2-3}{after 3}`

\_\_\_\_\_before 2\_\_\_\_\_

`\uncover`, `\visible`, `\only`, `\alt`, `\temporal`

current slide 2

sl.2: `\uncover<2-3>{uncovered on slides 2-3}`

\_\_\_\_\_uncovered on slides 2-3\_\_\_\_\_

sl.2: `\visible<2-3>{visible on slides 2-3}`

\_\_\_\_\_visible on slides 2-3\_\_\_\_\_

sl.2: `\only<2-3>{only on 2-3}`

\_\_\_\_\_only on 2-3\_\_\_\_\_

sl.2: \_\_\_\_\_ `\alt<2-3>{on 2-3}{not on 2-3}` \_\_\_\_\_

\_\_\_\_\_on 2-3\_\_\_\_\_

sl.2: `\temporal<2-3>{before 2}{on 2-3}{after 3}`

\_\_\_\_\_on 2-3\_\_\_\_\_

`\uncover`, `\visible`, `\only`, `\alt`, `\temporal`

current slide 3

sl.3: `\uncover<2-3>{uncovered on slides 2-3}`

\_\_\_\_\_uncovered on slides 2-3\_\_\_\_\_

sl.3: `\visible<2-3>{visible on slides 2-3}`

\_\_\_\_\_visible on slides 2-3\_\_\_\_\_

sl.3: `\only<2-3>{only on 2-3}`

\_\_\_\_\_only on 2-3\_\_\_\_\_

sl.3: \_\_\_\_\_`\alt<2-3>{on 2-3}{not on 2-3}`\_\_\_\_\_

\_\_\_\_\_on 2-3\_\_\_\_\_

sl.3: `\temporal<2-3>{before 2}{on 2-3}{after 3}`

\_\_\_\_\_on 2-3\_\_\_\_\_

`\uncover`, `\visible`, `\only`, `\alt`, `\temporal`

current slide 4

sl.4: `\uncover<2-3>{uncovered on slides 2-3}`

\_\_\_\_\_uncovered on slides 2-3\_\_\_\_\_

sl.4: `\visible<2-3>{visible on slides 2-3}`

\_\_\_\_\_

sl.4: `\only<2-3>{only on 2-3}`

\_\_\_\_\_

sl.4: \_\_\_\_\_ `\alt<2-3>{on 2-3}{not on 2-3}` \_\_\_\_\_

\_\_\_\_\_not on 2-3\_\_\_\_\_

sl.4: `\temporal<2-3>{before 2}{on 2-3}{after 3}`

\_\_\_\_\_after 3\_\_\_\_\_

## Kommando `\onslide`

hier nur Syntax für den einfachen Fall:

- ▶ `\onslide<<overlay spec>>` oder  
`\onslide+<<overlay spec>>`
- ▶ Unterschied wie zwischen `\uncover` und `\visible`
- ▶ alles nach diesem Kommando wird nur „gezeigt“, wenn `beamerpauses` einen Wert hat, der zu `<overlay spec>` passt

## Verallgemeinerung $\langle action\ spec \rangle$

- ▶ Doku: *action specifications* sind «*a rather advanced concept*».
- ▶ mögliche actions sind
  - ▶ `uncover`      der Default
  - ▶ `alert`
  - ▶ `only`
  - ▶ `visible`
  - ▶ `invisible`
- ▶ im folgenden nur ein einfaches Beispiel



## Kommando `\item`

- ▶ wichtiges Beispiel eines Kommandos, das von `beamer` „action spec aware“ gemacht wird
- ▶ betrachten/benutzen (fast) nur `uncover`, was nicht explizit notiert werden muss, da Default
- ▶ dann sieht eine  $\langle action\ spec \rangle$  aus wie eine  $\langle overlay\ spec \rangle$

Beispiel:

```
\begin{itemize}
\item<1-> eins
\item<2-> zwei
\item<3-> drei
\end{itemize}
```

- ▶ eins
- ▶ zwei
- ▶ drei

## Kommando `\item`

- ▶ wichtiges Beispiel eines Kommandos, das von `beamer` „action spec aware“ gemacht wird
- ▶ betrachten/benutzen (fast) nur `uncover`, was nicht explizit notiert werden muss, da Default
- ▶ dann sieht eine  $\langle action\ spec \rangle$  aus wie eine  $\langle overlay\ spec \rangle$

Beispiel:

```
\begin{itemize}
\item<1-> eins
\item<2-> zwei
\item<3-> drei
\end{itemize}
```

- ▶ eins
- ▶ zwei
- ▶ drei

## Kommando `\item`

- ▶ wichtiges Beispiel eines Kommandos, das von `beamer` „action spec aware“ gemacht wird
- ▶ betrachten/benutzen (fast) nur `uncover`, was nicht explizit notiert werden muss, da Default
- ▶ dann sieht eine  $\langle action\ spec \rangle$  aus wie eine  $\langle overlay\ spec \rangle$

Beispiel:

```
\begin{itemize}
\item<1-> eins
\item<2-> zwei
\item<3-> drei
\end{itemize}
```

- ▶ eins
- ▶ zwei
- ▶ drei

## Kommando `\item`: inkrementelle Spezifikationen

Beispiel:

```
\begin{itemize}
\item<+> zwei
\item<+> zweieinhalb
\item<+> drei
\end{itemize}
```

- ▶ zwei
- ▶ zweieinhalb
- ▶ drei

etwas kürzer:

```
\begin{itemize}[<+>]
\item zwei
\item zweieinhalb
\item drei
\end{itemize}
```

- ▶ zwei
- ▶ zweieinhalb
- ▶ drei

## Kommando `\item`: inkrementelle Spezifikationen

Beispiel:

```
\begin{itemize}
\item<+> zwei
\item<+> zweieinhalb
\item<+> drei
\end{itemize}
```

- ▶ zwei
- ▶ zweieinhalb
- ▶ drei

etwas kürzer:

```
\begin{itemize}[<+>]
\item zwei
\item zweieinhalb
\item drei
\end{itemize}
```

- ▶ zwei
- ▶ zweieinhalb
- ▶ drei

## Kommando `\item`: inkrementelle Spezifikationen

Beispiel:

```
\begin{itemize}
```

```
\item<+> zwei
```

▶ zwei

```
\item<+> zweieinhalb
```

▶ zweieinhalb

```
\item<+> drei
```

▶ drei

```
\end{itemize}
```

etwas kürzer:

```
\begin{itemize}[<+>]
```

```
\item zwei
```

▶ zwei

```
\item zweieinhalb
```

▶ zweieinhalb

```
\item drei
```

▶ drei

```
\end{itemize}
```

## Kommando `\item`: inkrementelle Spezifikationen

Beispiel:

```
\begin{itemize}
\item<+> zwei
\item<+> zweieinhalb
\item<+> drei
\end{itemize}
```

- ▶ zwei
- ▶ zweieinhalb
- ▶ drei

etwas kürzer:

```
\begin{itemize}[<+>]
\item zwei
\item zweieinhalb
\item drei
\end{itemize}
```

- ▶ zwei
- ▶ zweieinhalb
- ▶ drei

## Kommando `\item`: inkrementelle Spezifikationen

Beispiel:

```
\begin{itemize}
\item<+> zwei
\item<+> zweieinhalb
\item<+> drei
\end{itemize}
```

- ▶ zwei
- ▶ zweieinhalb
- ▶ drei

etwas kürzer:

```
\begin{itemize}[<+>]
\item zwei
\item zweieinhalb
\item drei
\end{itemize}
```

- ▶ zwei
- ▶ zweieinhalb
- ▶ drei



## Kommando `\item`: inkrementelle Spezifikationen

Beispiel:

```
\begin{itemize}
\item<+> zwei
\item<+> zweieinhalb
\item<+> drei
\end{itemize}
```

- ▶ zwei
- ▶ zweieinhalb
- ▶ drei

etwas kürzer:

```
\begin{itemize}[<+>]
\item zwei
\item zweieinhalb
\item drei
\end{itemize}
```

- ▶ zwei
- ▶ zweieinhalb
- ▶ drei

## Inkrementelle Spezifikationen (2)

- ▶ Bedeutung des `+`
  1. der aktuelle Wert von `beamerpauses` wird eingesetzt
  2. `beamerpauses` wird um 1 erhöht
- ▶ statt `+` ist auch `+(<diff>)` möglich
  - ▶ das habe ich noch nie gebraucht
  
- ▶ es gibt auch noch `.`
  1. der Zähler `beamerpauses` wird *nicht* verändert
  2. der Wert von `beamerpauses` *minus* 1 wird geliefert

## Inkrementelle Spezifikationen (3): Beispiel

aus der `beamer`-Dokumentation:

```
\begin{enumerate}[<+>]
\item This is \alert<.>{important}.
\item highlight \alert<.>{this} and \alert<.>{this}
\item What is the \alert<.>{matrix}?
\end{enumerate}
```

1. This is **important**.
2. highlight this and this
3. What is the matrix?

## Inkrementelle Spezifikationen (3): Beispiel

aus der `beamer`-Dokumentation:

```
\begin{enumerate}[<+>]
\item This is \alert<.>{important}.
\item highlight \alert<.>{this} and \alert<.>{this}
\item What is the \alert<.>{matrix}?
\end{enumerate}
```

1. This is important.
2. highlight **this** and **this**
3. What is the matrix?

## Inkrementelle Spezifikationen (3): Beispiel

aus der `beamer`-Dokumentation:

```
\begin{enumerate}[<+>]
\item This is \alert<.>{important}.
\item highlight \alert<.>{this} and \alert<.>{this}
\item What is the \alert<.>{matrix}?
\end{enumerate}
```

1. This is important.
2. highlight this and this
3. What is the **matrix**?

# Überblick

Lokale dynamische Struktur: Overlays

Globale dynamische Struktur: „Hin- und Her-Springen“





Hypertargets und Hyperlinks

„Hineinzoomen“ in Bilder

## Zeitplanung für einen Vortrag

- ▶ weniger als 2 Minuten pro Folie sind meist illusorisch
- ▶ *laut* üben
- ▶ das Überspringen von Vortragsteilen einplanen

## Targets, Links und Buttons

- ▶ `\hypertarget<⟨overlay spec⟩>{⟨target name⟩}{⟨text⟩}`
  - ▶ `⟨text⟩` ist auf allen Slides
  - ▶ auf Slide `⟨overlay spec⟩` ist er auch Sprungziel mit Namen `⟨target name⟩`
- ▶ `\hyperlink{⟨target name⟩}{⟨link text⟩}`
- ▶ beamer buttons
  - ▶ `\beamerbutton{⟨text⟩}` 
  - ▶ `\beamergotobutton{⟨text⟩}` 
  - ▶ `\beamerreturnbutton{⟨text⟩}` 
  - ▶ `\beamerskipbutton{⟨text⟩}` 
- ▶ (siehe auch: die Folien zum Kapitel „Verweise“)

`\hyperlink{hauptsatz}{%`

`\beamergotobutton{Hauptsatz der Präsentationstechnik}}`

hier geht es zum 



## Vergrößerung von Bildern

- ▶ im laufenden „Text“

```
\begin{frame}<1>[label=grossesbild]
```

```
...
```

```
<Bild>
```

```
...
```

```
\framezoom<1><2>(0mm,-2mm)(42mm,23mm)
```

```
\framezoom<1><3>(0mm,36mm)(32mm,20mm)
```

```
\framezoom<1><4>(72mm,17mm)(30mm,23mm)
```

```
\end{frame}
```

- ▶ (z. B.) im Anhang

```
\againframe<2->[plain]{grossesbild}
```

# Überblick

## Der Hauptsatz der Präsentationstechnik

# Hauptsatz der Präsentationstechnik

Beispiel einer `theorem` Umgebung

```
\begin{theorem}[\hypertarget<1>{hauptsatz}{Hauptsatz  
der Präsentationstechnik}]
```

Für ... lang.

```
\end{theorem}
```

liefert

Theorem (Hauptsatz der Präsentationstechnik)

*Für jede nichtnegative ganze Zahl  $n$  gilt:*

*Wenn  $n \geq 1$ , dann ist ein Vortrag mit  $n$  Folien zu lang.*

# Beweis des Hauptsatzes

## Schritt 1: der Plan

vollständige Induktion — was sonst

# Beweis des Hauptsatzes

## Schritt 2: Induktionsanfang

$$n = 0$$

**Zeige:** Wenn  $0 \geq 1$ , dann ist ein Vortrag mit 0 Folien zu lang.

**Beweis:** trivialerweise wahr, da  $0 \geq 1$  falsch ist.

# Beweis des Hauptsatzes

## Schritt 3: Induktionsvoraussetzung

für ein beliebiges aber festes  $n$  gelte:

Wenn  $n \geq 1$ , dann ist ein Vortrag mit  $n$  Folien zu lang.

## Beweis des Hauptsatzes

### Schritt 4: Induktionsschluss

**Zeige:** Wenn  $n + 1 \geq 1$ , dann ist ein Vortrag mit  $n + 1$  Folien zu lang.

- Beweis:**
- ▶ Nach Induktionsvoraussetzung sind  $n$  Folien schon zu viel.
  - ▶  $n + 1$  Folien sind mehr als  $n$  Folien.
  - ▶ Also sind  $n + 1$  Folien auch zu viel.

# Beweis des Hauptsatzes

## Schritt 5: Fehlersuche

*irgendwas stimmte da nicht ...*

```
\hyperlink{zurueckvonhauptsatz}{%  
\beamerreturnbutton{zurück zum Thema}}
```

◀ zurück zum Thema