

# Patterns Generate Architectures

en artikel af Kent Beck og Ralph Johnson

Præsentation af Johannes Serup

8. februar 2008

# Introduktion

Begreber

Inspiration

Formål

Eksempel: HotDraw

# Begreber

- ▶ Arkitektur. Hvordan de enkelte dele arbejder sammen i et komplet system.

# Begreber

- ▶ Arkitektur. Hvordan de enkelte dele arbejder sammen i et komplet system.
- ▶ Pattern. Beskrivelse af dele af arkitekturen.

# Begreber

- ▶ Arkitektur. Hvordan de enkelte dele arbejder sammen i et komplet system.
- ▶ Pattern. Beskrivelse af dele af arkitekturen.
- ▶ Framework. De dele af systemet der kan genbruges. En konkret måde at repræsentere en arkitektur.

# Inspiration



- ▶ Arkitekt/Matematiker Chrisopher Alexander.

# Inspiration



- ▶ Arkitekt/Matematiker Chrisopher Alexander.
- ▶ *Hvorfor og hvornår* et pattern bør anvendes (Generative Patterns).

# Inspiration



- ▶ Arkitekt/Matematiker Chrisopher Alexander.
- ▶ *Hvorfor og hvornår* et pattern bør anvendes (Generative Patterns).
- ▶ Ikke nødvendigt at være ekspert.



# Inspiration



- ▶ Arkitekt/Matematiker Chrisopher Alexander.
- ▶ *Hvorfor og hvornår* et pattern bør anvendes (Generative Patterns).
- ▶ Ikke nødvendigigt at være ekspert.
- ▶ Indfører et fælles sprog.

# Formål

- ▶ Hvordan arkitekturer udledes fra en sekvens af patterns.

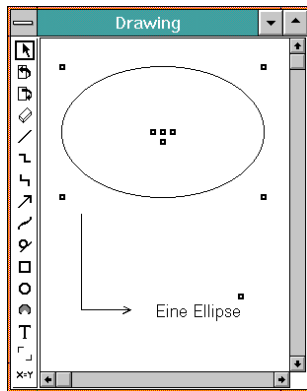
# Formål

- ▶ Hvordan arkitekturer udledes fra en sekvens af patterns.
- ▶ Give et overblik over beslutninger undervejs.

# Formål

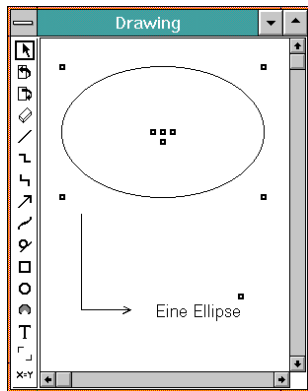
- ▶ Hvordan arkitekturer udledes fra en sekvens af patterns.
- ▶ Give et overblik over beslutninger undervejs.
- ▶ Opnå en forståelse for baggrunden af disse valg.

# Eksempel: HotDraw



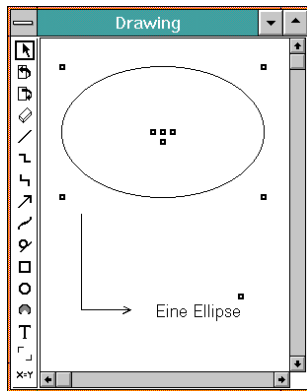
- Realisere et basalt tegneprogram.

## Eksempel: HotDraw



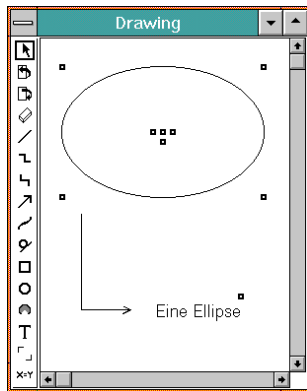
- ▶ Realisere et basalt tegneprogram.
- ▶ Analysere de enkelte behov/funktioner.

## Eksempel: HotDraw



- ▶ Realisere et basalt tegneprogram.
- ▶ Analysere de enkelte behov/funktioner.
- ▶ Hvilke patterns kan bruges?

## Eksempel: HotDraw



- ▶ Realisere et basalt tegneprogram.
- ▶ Analysere de enkelte behov/funktioner.
- ▶ Hvilke patterns kan bruges?
- ▶ Udled den endelige arkitektur.



## Eksempel: HotDraw - Tjekliste

- ▶ 2 typer patterns: Objekt-Orienterede patterns og Grafik patterns.

## Eksempel: HotDraw - Tjekliste

- ▶ 2 typer patterns: Objekt-Orienterede patterns og Grafik patterns.
- ▶ Preconditions: Forudsætning for at anvende det konkrete pattern.

## Eksempel: HotDraw - Tjekliste

- ▶ 2 typer patterns: Objekt-Orienterede patterns og Grafik patterns.
- ▶ Preconditions: Forudsætning for at anvende det konkrete pattern.
- ▶ Problem: Hvilken type problem der adresseres.

## Eksempel: HotDraw - Tjekliste

- ▶ 2 typer patterns: Objekt-Orienterede patterns og Grafik patterns.
- ▶ Preconditions: Forudsætning for at anvende det konkrete pattern.
- ▶ Problem: Hvilken type problem der adresseres.
- ▶ Constraints: Konflikter og tradeoffs (tid vs kompleksitet).

## Eksempel: HotDraw - Tjekliste

- ▶ 2 typer patterns: Objekt-Orienterede patterns og Grafik patterns.
- ▶ Preconditions: Forudsætning for at anvende det konkrete pattern.
- ▶ Problem: Hvilken type problem der adresseres.
- ▶ Constraints: Konflikter og tradeoffs (tid vs kompleksitet).
- ▶ Solution: Opsummering af løsning på problemet.

# Eksempel: HotDraw - MVC

- ▶ User-interface (Model-View-Controller).

## Eksempel: HotDraw - MVC

- ▶ User-interface (Model-View-Controller).
- ▶ Drawing, DrawingView og DrawingController.

## Eksempel: HotDraw - MVC

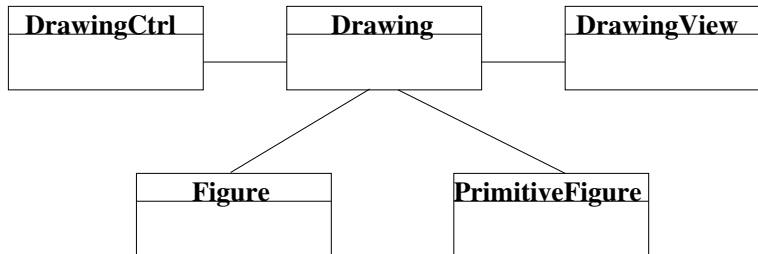
- ▶ User-interface (Model-View-Controller).
- ▶ Drawing, DrawingView og DrawingController.
- ▶ Flere elementer: anvend **Composite** på Drawing.



## Eksempel: HotDraw - MVC

- ▶ User-interface (Model-View-Controller).
- ▶ Drawing, DrawingView og DrawingController.
- ▶ Flere elementer: anvend **Composite** på Drawing.
- ▶ Resultatet er objekterne Figure og PrimitiveFigure.

## Eksempel: HotDraw - MVC (diagram)



## Eksempel: HotDraw - Behandling af input

- ▶ Tilstanden for et objekt `Tool` afhænger af brugerens valg fra menuen.

## Eksempel: HotDraw - Behandling af input

- ▶ Tilstanden for et objekt `Tool` afhænger af brugerens valg fra menuen.
- ▶ Anvend **Objects of States** på `DrawingController`.

## Eksempel: HotDraw - Behandling af input

- ▶ Tilstanden for et objekt Tool afhænger af brugerens valg fra menuen.
- ▶ Anvend **Objects of States** på DrawingController.
- ▶ DrawingController opdatere Drawing vha Tool.

## Eksempel: HotDraw - Behandling af input

- ▶ Tilstanden for et objekt Tool afhænger af brugerens valg fra menuen.
- ▶ Anvend **Objects of States** på DrawingController.
- ▶ DrawingController opdatere Drawing vha Tool.
- ▶ Forskellige billeder kræver forskellige Tool objekter. Anvend **Editor**.

# Eksempel: HotDraw - Synkronisering af objekter

- ▶ Når Figure ændres skal skærmen opdateres.

## Eksempel: HotDraw - Synkronisering af objekter

- ▶ Når Figure ændres skal skærmen opdateres.
- ▶ Anvend **Observer** på Figure.



## Eksempel: HotDraw - Lokale opdateringer i User speed

- ▶ Opdatering af display, men kun der hvor der er foretaget ændringer.

## Eksempel: HotDraw - Lokale opdateringer i User speed

- ▶ Opdatering af display, men kun der hvor der er foretaget ændringer.
- ▶ Anvend **Collect Damage** på `DrawingView`.

## Eksempel: HotDraw - Lokale opdateringer i User speed

- ▶ Opdatering af display, men kun der hvor der er foretaget ændringer.
- ▶ Anvend **Collect Damage** på `DrawingView`.
- ▶ Opdater display hver gang en event er udført.

## Eksempel: HotDraw - Lokale opdateringer i User speed

- ▶ Opdatering af display, men kun der hvor der er foretaget ændringer.
- ▶ Anvend **Collect Damage** på `DrawingView`.
- ▶ Opdater display hver gang en event er udført.
- ▶ Anvend **Update at User Speed** på `DrawingController`.

## Eksempel: HotDraw - Handles

- ▶ Hver Figure har et Handle objekt.

## Eksempel: HotDraw - Handles

- ▶ Hver Figure har et Handle objekt.
- ▶ Introducer et objekt til at håndtere Handle objekter

## Eksempel: HotDraw - Handles

- ▶ Hver Figure har et Handle objekt.
- ▶ Introducer et objekt til at håndtere Handle objekter
- ▶ Anvend **Wrapper/Adaptor** på Drawing.

## Eksempel: HotDraw - Handles

- ▶ Hver Figure har et Handle objekt.
- ▶ Introducer et objekt til at håndtere Handle objekter
- ▶ Anvend **Wrapper/Adaptor** på Drawing.
- ▶ Resultere i et nyt objekt SelectionDrawing som holder styr på Handle objekter.



## Opsummering

- ▶ Følgende 7 patterns er brugt til at udlede HotDraw:

# Opsummering

- ▶ Følgende 7 patterns er brugt til at udlede HotDraw:
  - ▶ **Composite**
  - ▶ **Objects of States**
  - ▶ **Editor**
  - ▶ **Observer**
  - ▶ **Collect Damage**
  - ▶ **Update at User Speed**
  - ▶ **Wrapper/Adaptor**

## Opsummering

- ▶ Følgende 7 patterns er brugt til at udlede HotDraw:
  - ▶ **Composite**
  - ▶ **Objects of States**
  - ▶ **Editor**
  - ▶ **Observer**
  - ▶ **Collect Damage**
  - ▶ **Update at User Speed**
  - ▶ **Wrapper/Adaptor**
- ▶ Begrund valg i hvert skridt.

## Opsummering

- ▶ Følgende 7 patterns er brugt til at udlede HotDraw:
  - ▶ **Composite**
  - ▶ **Objects of States**
  - ▶ **Editor**
  - ▶ **Observer**
  - ▶ **Collect Damage**
  - ▶ **Update at User Speed**
  - ▶ **Wrapper/Adaptor**
- ▶ Begrund valg i hvert skridt.
- ▶ Når der ikke kan tilføjes flere patterns er arkitekturen udledt.

# Opsummering

- ▶ Følgende 7 patterns er brugt til at udlede HotDraw:
  - ▶ **Composite**
  - ▶ **Objects of States**
  - ▶ **Editor**
  - ▶ **Observer**
  - ▶ **Collect Damage**
  - ▶ **Update at User Speed**
  - ▶ **Wrapper/Adaptor**
- ▶ Begrund valg i hvert skridt.
- ▶ Når der ikke kan tilføjes flere patterns er arkitekturen udledt.
- ▶ Den udledte arkitektur kan genbruges eller dele (patterns) kan nemt udskiftes.