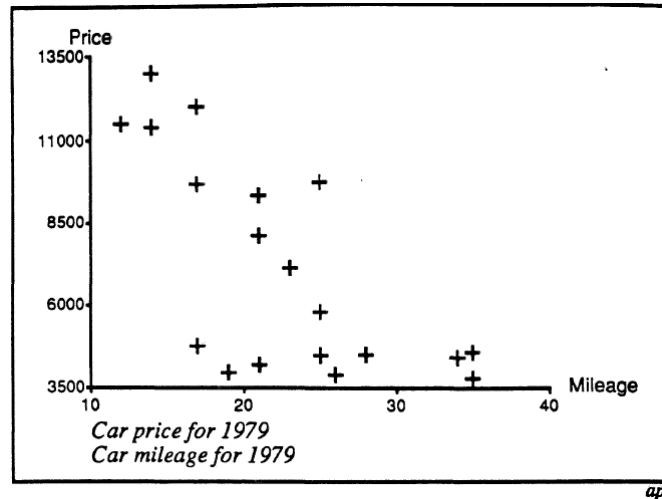# Smart Graphics: Graphik und Kommunikation

Vorlesung „Smart Graphics"

Andreas Butz

Graphik und Kommunikation

- Eine „Grammatik" für Diagramme
  - Syntax
  - Semantik
- Rhetoric Structure Theory
  - Grundidee
  - Anwendung auf multimediale Präsentationen

Eine Grammatik für Diagramme: APT
[Mackinlay 1986]

Price
13500 -
11000 -
8500 -
6000 -
3500 -
10   20   30   40   Mileage

Car price for 1979
Car mileage for 1979

apt

Ausgangsdaten

Price(Accord,5799)   *Price*(AMC Pacer,4749)
Mileage(Accord,25)   *Mileage*(AMC Pacer,17)
Weight(Accord,2240)   *Weight*(AMC Pacer,3350)
Repair(Accord,Great)   *Repair*(AMC Pacer,Terrible)
Nation(Accord,Germany)   *Nation*(AMC Pacer,USA)
Price(Audi 5000,9690)   *Price*(BMW 320i,9735)
              ⋮   ⋮

Figure 2: Relation Tuples About 1979 Automobiles. *This is an example of a table of relation tuples that might be generated by a database system in response to a query. A presentation tool can do much better than this.*

Price :   *Cars* → [3500,13000]
Mileage :   *Cars* → [10,40]
Weight :   *Cars* → [1500,5000]
Repair :   *Cars* → ⟨Great,Good,OK,Bad,Terrible⟩
Nation :   *Cars* → {USA,Germany,France,···}
Cars = {Accord,AMC Pacer,Audi 5000,BMW 320i,···}

Figure 3: Structural Properties of the Automobile Relations. *The arrow (→) indicates a functional dependency between domain sets. The square brackets ([]) describe domain sets that are quantitative ranges, the angle brackets (⟨⟩) describe domain sets that are ordered sets, and the curly braces ({}) describe domain sets that are unordered sets.*

Diagramme als visuelle Sprache

- Graphische Elemente (Punkte, Linien, ..) sind die „Worte"
- Mehrere Elemente werden zu „Sätzen" kombiniert (Syntax)
- Worte und Sätze haben Bedeutungen (Semantik)
- Visualisierung = Generierung eines Satzes, der genau die beabsichtigten Inhalte ausdrückt

A set of facts is *expressible* in a language if it contains a sentence that:

    1) encodes all the facts in the set and

    2) encodes only the facts in the set.

Formale Beschreibung eines Diagrams

$Encodes(VertAxis, [3500, 13000], ScatterPlot)$
$Encodes(HorzAxis, [10, 40], ScatterPlot)$
$Encodes(Points, Cars, ScatterPlot)$
$Encodes(Position(Points, VertAxis), Price(Cars), ScatterPlot)$
$Encodes(Position(Points, HorzAxis), Mileage(Cars), ScatterPlot)$

Figure 4: The Graphical Design for a Scatter Plot of the Price/Mileage Input. *The Encodes relation indicates the relationship between graphical objects or properties and the information encoded. For example, the first line says that the vertical axis encodes the range of prices, and the fourth line says that the position of the points on the vertical axis encodes the prices of cars. The input relations are written as functions to simplify the description.*
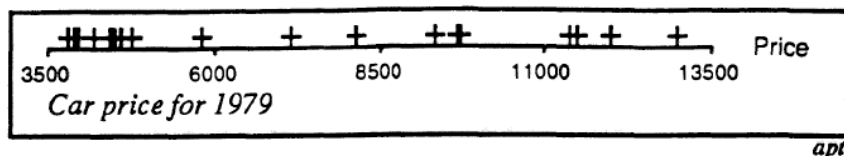

Figure 9: The Horizontal Position Sentence of the Price Relation.
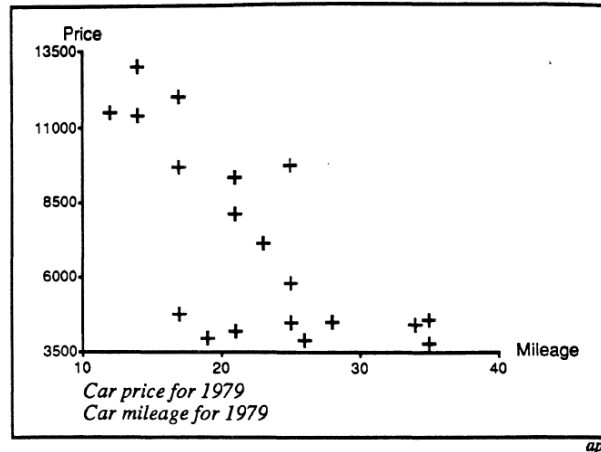
Zugehörige graphische Ausgabe

Figure 5: Scatter Plot of the Price/Mileage Input. *The graphical design for this image is in Figure 4. The design expresses the relations only if the application permits the details about the cars to be omitted. The "apt" in the lower right corner indicates that APT designed and rendered this diagram.*
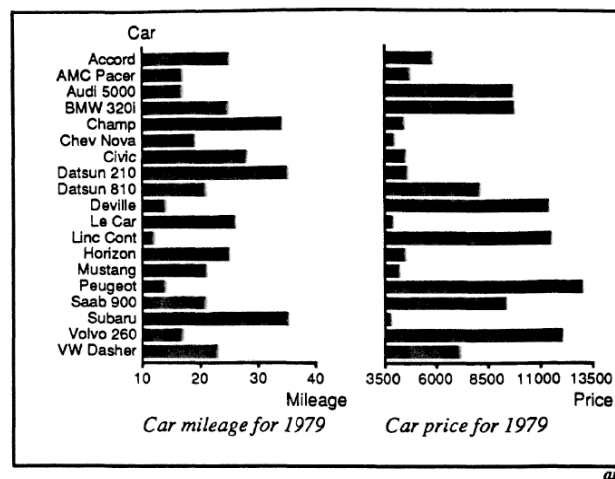
Alternative graphische Umsetzung

Figure 7: Aligned Bar Chart for the Price/Mileage Input. *This diagram shows the detailed properties of the cars better than a scatter plot. However, the general relationships are not as easy to see.*
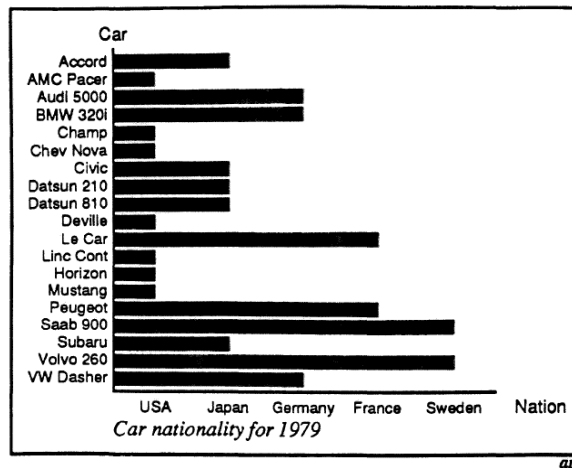
Gegenbeispiel

Figure 11: Incorrect Use of a Bar Chart for the Nation Relation. *The lengths of the bars suggest an ordering on the vertical axis, as if the USA cars were longer or better than the other cars, which is not true for the Nation relation.*

Rhetoric Structure Theory [Mann & Thompson, 1988]

- Beschreibung der hierarchischen Struktur eines Textes
- Unterteilung des Textes in logische Einheiten, z.B. Teilsätze
- Relationen zwischen diesen Einheiten
  - Nukleus – Relation – Satellit
- Insgesamt 23 Relationen
  - Preparation, elaboration, background, purpose,…
- Schemata für typische Kombinationen
- Fast jeder Text kann durch einen RST-Baum beschrieben werden

RST - Beispiel
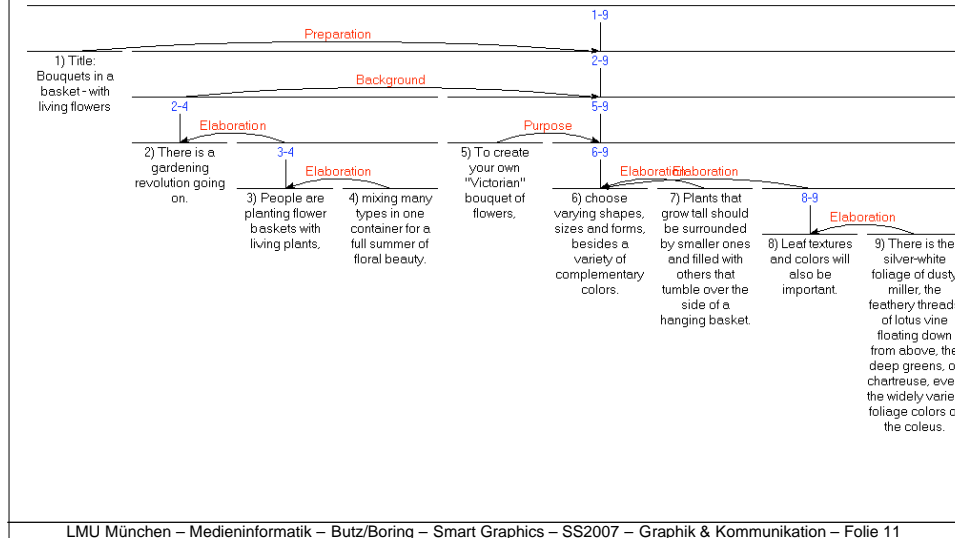
# RST Multimedia Planung in WIP
[Andre & Rist 1997]

- Verallgemeinerung von RST auf Multimediale Inhalte
  - Text, Bilder, Animationen, (Zeigegesten)
- Planung multimedialer Dokumente als hierarchischer Planungsprozess
  - Beginne mit der Wurzel des RST-Baumes
  - Unterteile in logische Einheiten
  - Lege Medium für jede Einheit fest
  - Generiere jeweilige Einheit (Text, Bild, Animation)

WIP Architektur
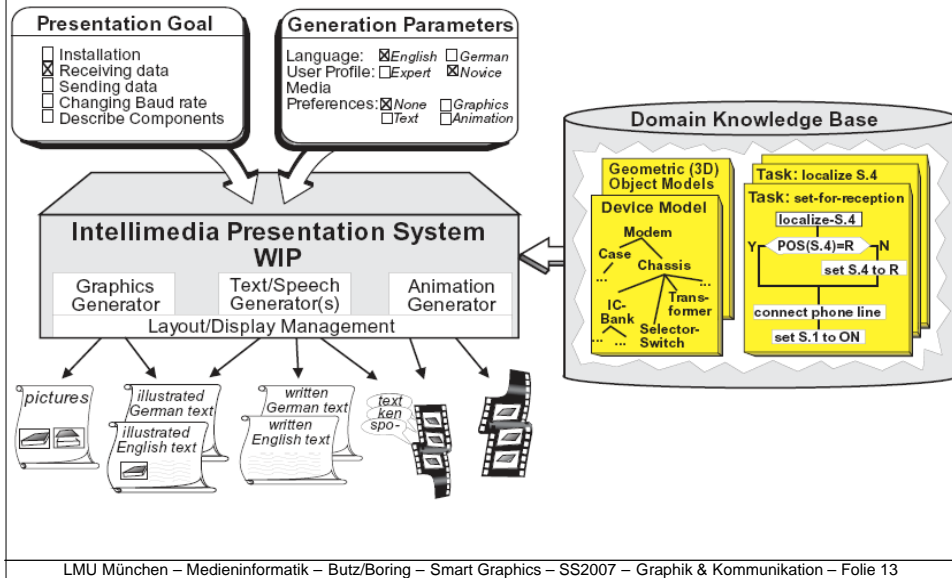
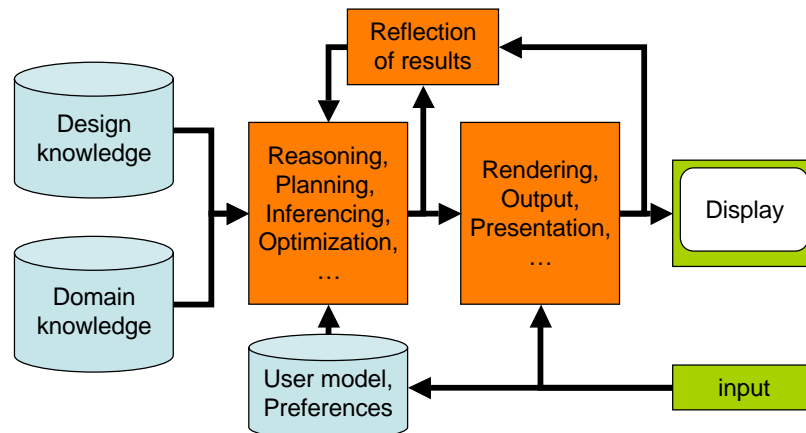# WIP Generierungsbeispiel
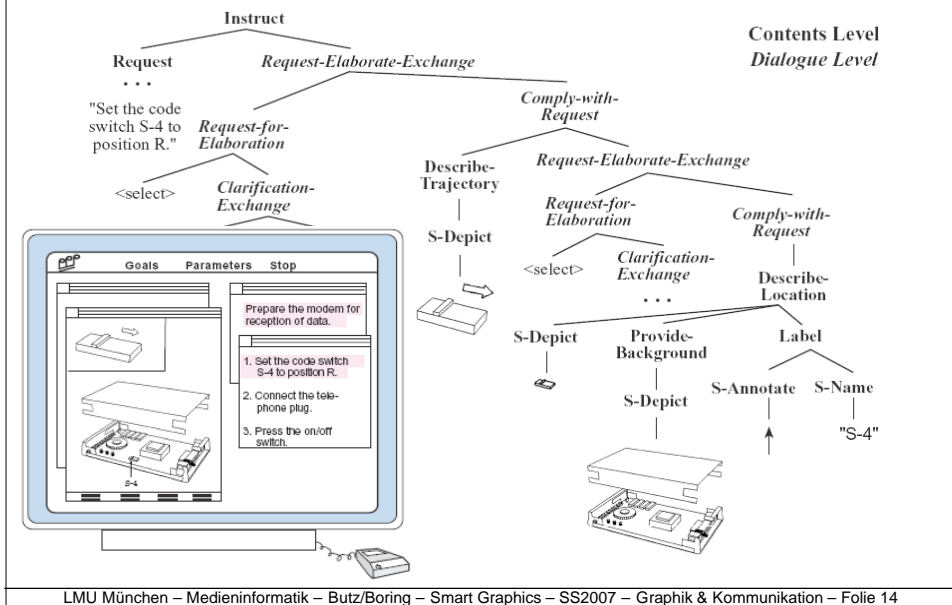
Smart Graphics: Methoden 1

Themen heute

- Generisches Modell eines SG systems
- Praktisches Beispiel: Generierung von 3D-Animationen
- Dabei insbesondere: hierarchische Planung

Some typical elements of SG systems

- Strong simplification and generalization
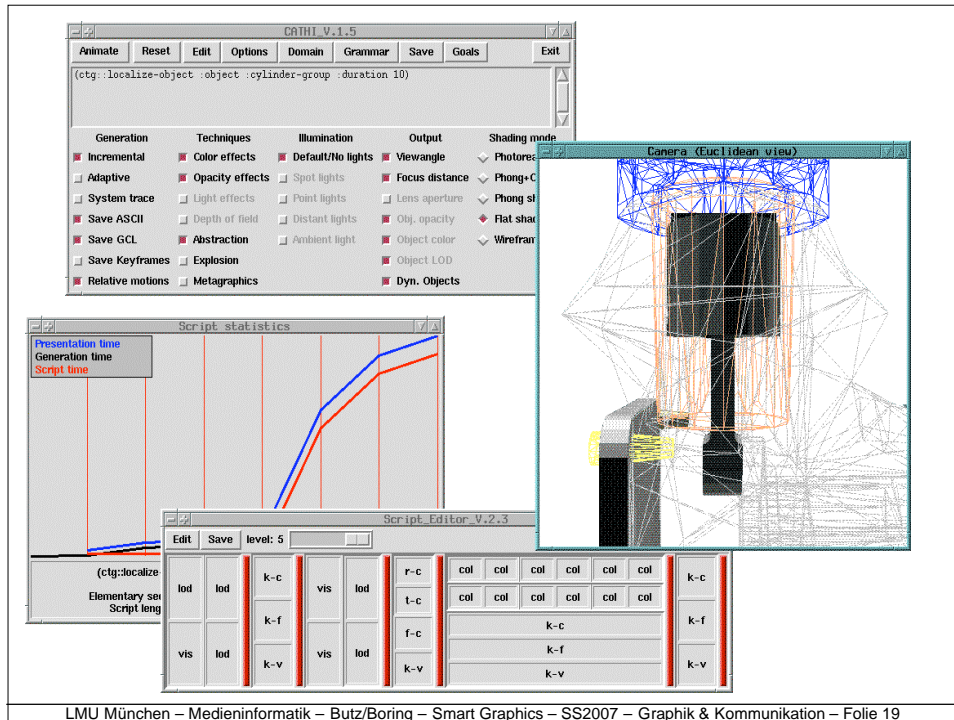- Often only some elements present
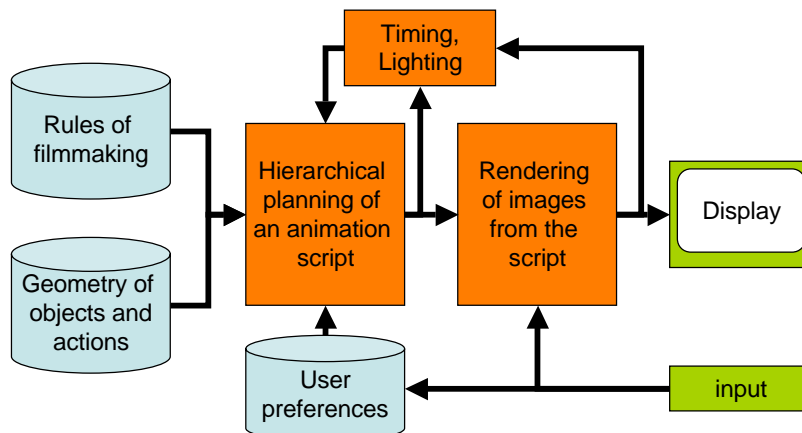
Concrete example: filmmaking

- Task: create a 3D animation for the explanation of a technical device
- Starting point: communicative goal
  - Example: show where the switch X is
- Intended result: 3D animation
 - E.g., showing where switch X is

Concrete example: filmmaking

- Example system CATHI [Butz, 97]

Knowledge representation

- Representations can only capture part of the reality
  - Which aspects do we need to model?
  - At which level of detail do we need to model them?
  - Do we need qualitative or quantitative knowledge?
  - How do we want to process the knowledge?
- Different kinds of knowledge must be represented
  - Domain knowledge
  - Design knowledge
  - Knowledge about the user

Domain knowledge
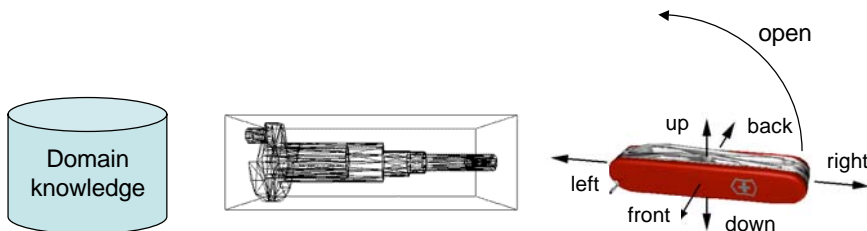
Design knowledge

User model, Preferences
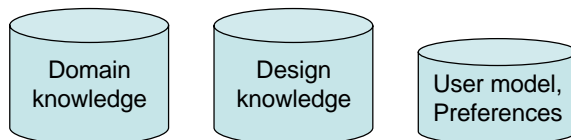
Domain Knowledge

- Knowl. about things in the problem domain, e.g.,
  - Road network in a geographic database
  - Personal picture or music collection with metadata
  - Text and picture blocks for a magazine page
- Exchangeable if clearly separated from the rest
  - E.g., visualizations of different music collections
  - Route instructions in different cities
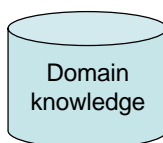
Domain knowledge

Domain Knowledge

- In the filmmaking example:
  - Geometries of objects + bounding boxes
  - Surfaces/colors of objects
  - Object groups and hierarchy
  - Preferred viewing directions of objects/groups
  - Trajectories of movements

open

up   back

left   right

front   down

Domain knowledge

---

Design Knowledge

- Knowledge how to structure graph. presentations, e.g.,
  - Rules of grid-based layout
  - Rules about the composition of an image
  - Rules about the composition of diagrams
- In the filmmaking example:
  - Formal „grammar" of the film language
  - Rules about temporal and spatial compositions of shots
- Must be formal enough to be used by a machine!
- When exchanged, changes visual style
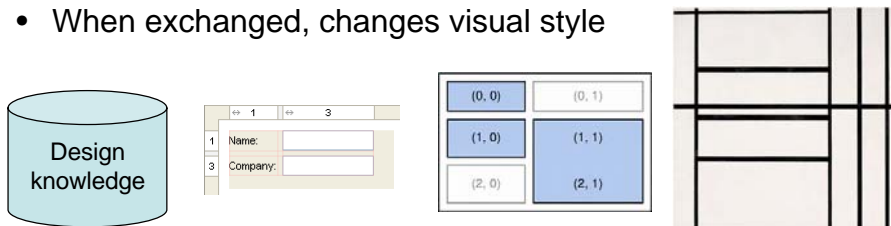
Design knowledge

⇔ 1	⇔ 3
1	Name:
3	Company:

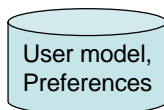| (0, 0)	(0, 1)
(1, 0)	(1, 1)
(2, 0)	(2, 1)

Example from CATHI:
a formal grammar of the film language

- Rules for decomposing sequences into subsequences
- Reusable in different situations
- Querying calculations in the 3D model
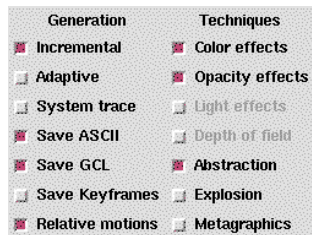- Details later
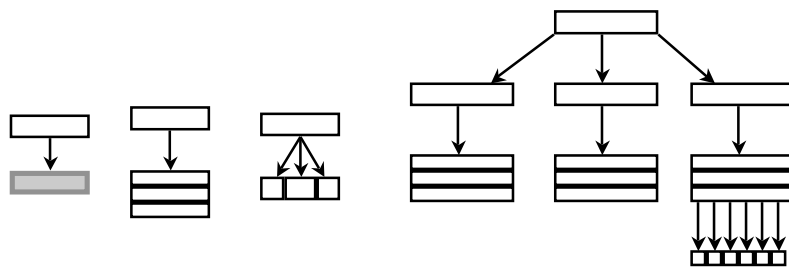
---

User model / preferences

- Knowledge about the user
  - Properties, such as level of expertise
  - Preferences, given implicitly or explicitly
  - Current context of the user
  - Also: capabilities of the output medium
- Examples
  - Previously bought items
  - Personal viewing preferences
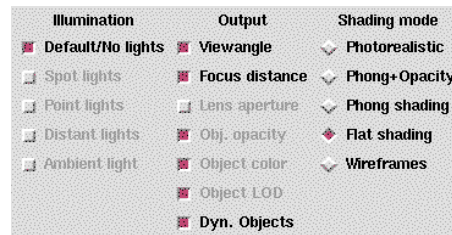 - Current resolution of the output screen

User model, Preferences

User preferences in CATHI

Generation	Techniques
▨ Incremental	▨ Color effects
▢ Adaptive	▨ Opacity effects
▢ System trace	▢ Light effects
▨ Save ASCII	▢ Depth of field
▨ Save GCL	▨ Abstraction
▢ Save Keyframes	▢ Explosion
▨ Relative motions	▢ Metagraphics

| Illumination	Output	Shading mode
▨ Default/No lights	▨ Viewangle	◇ Photorealistic
▢ Spot lights	▨ Focus distance	◇ Phong+Opacity
▢ Point lights	▢ Lens aperture	◇ Phong shading
▢ Distant lights	▨ Obj. opacity	◆ Flat shading
▢ Ambient light	▢ Object color	◇ Wireframes
	▨ Object LOD	
	▨ Dyn. Objects	

Stylistic preferences
of the user

Graphical capabilities of
the user's machine
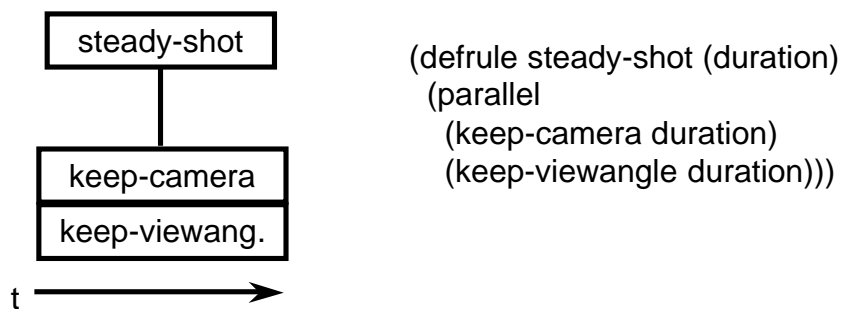(back in 1997!)

Reasoning

- Algorithms for:
  - Further refining the domain knowledge
  - Application of design knowledge
- Output of the reasoning process:
  - Complete structural description of the presentation
- Examples:
  - Route calculation on a road network
  - Layout of labels in a map
  - Layout of text blocks on a page
  - Specification of diagram elements
- Often the core of a SG system
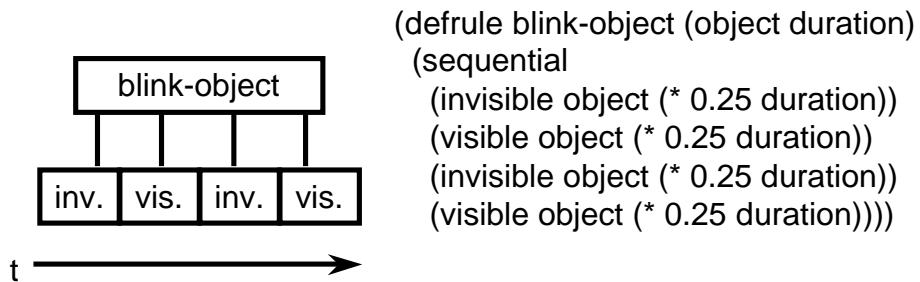
> Reasoning,
> Planning,
> Inferencing,
> Optimization,
> …

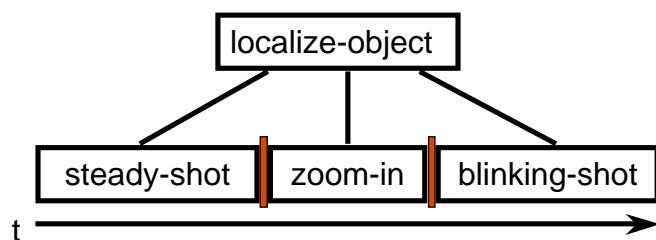# Example: animation scripts
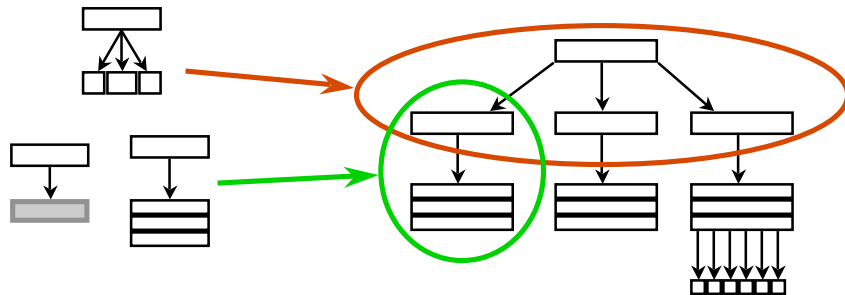
Parallel Decomposition


```
(defrule steady-shot (duration)
  (parallel
    (keep-camera duration)
    (keep-viewangle duration)))
```

Sequential Decomposition

(defrule blink-object (object duration)
  (sequential
    (invisible object (* 0.25 duration))
    (visible object (* 0.25 duration))
    (invisible object (* 0.25 duration))
    (visible object (* 0.25 duration))))
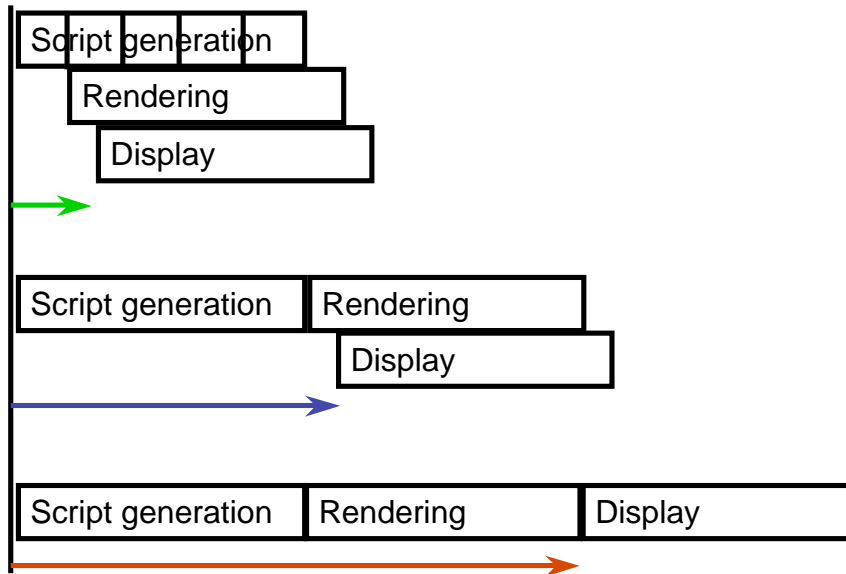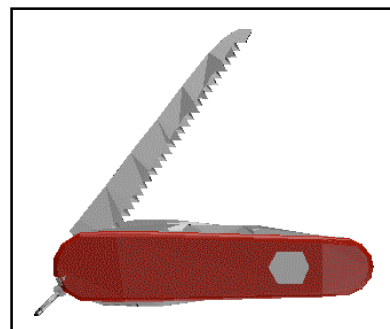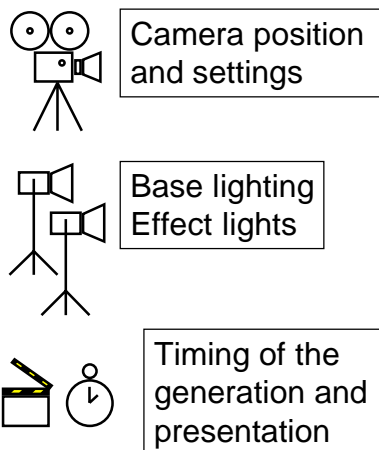
Incremental Decomposition

(defrule localize-object (object duration)
  (incremental
    (steady-shot (* 0.2 duration))
    (zoom-in object (* 0.4 duration))
    (blinking-shot object (* 0.4 duration))))

Why incremental generation?

Script generation

Rendering

Display

Script generation   Rendering

Display

Script generation   Rendering   Display

Current generation context

Camera position and settings

Base lighting Effect lights

Timing of the generation and presentation
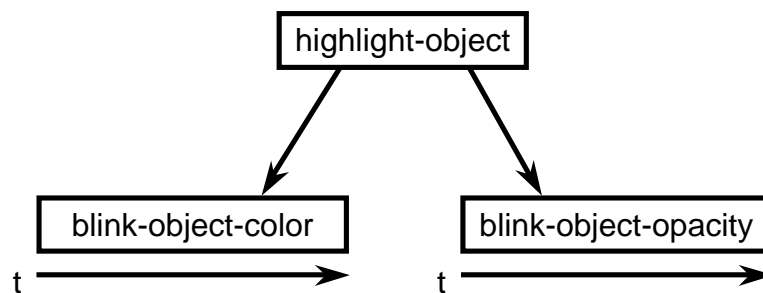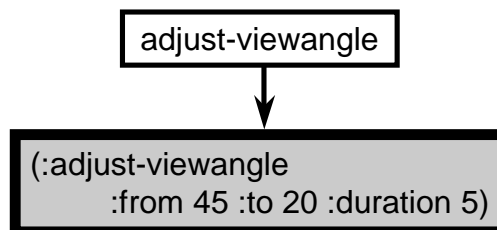
Object positions and properties

Conditional Decomposition

```
(defrule highlight-object (object duration)
  (if (feature color)
      (blink-object-color object duration)
      (blink-object-opacity object duration)))
```
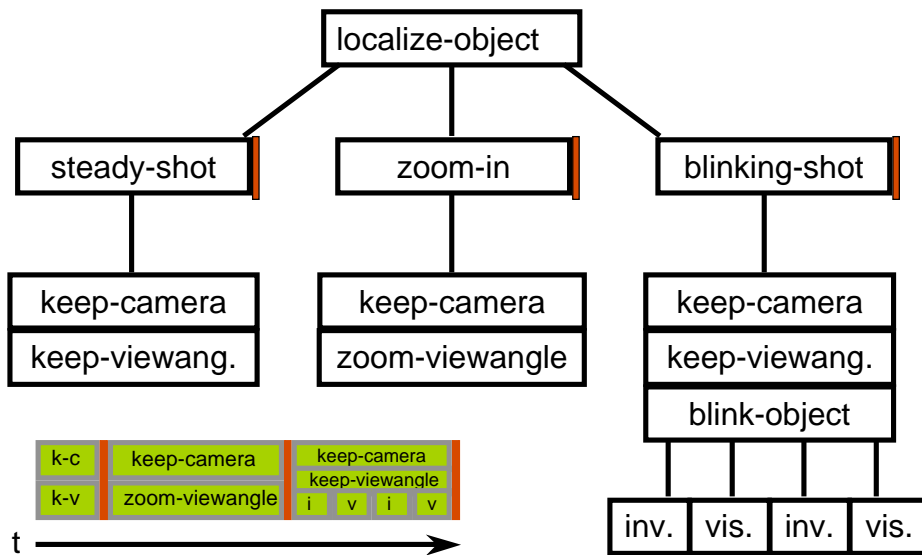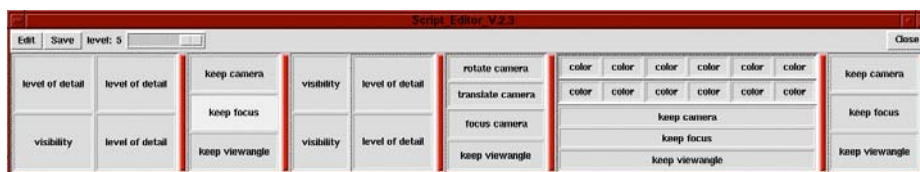
Translation of elementary sequences

```
(defrule adjust-viewangle (from to duration)
  `(:adjust-viewangle
      :from ,from :to ,to
      :duration ,duration))
```

Generierung eines Skripts

```
                    ┌──────────────────┐
                    │  localize-object │
                    └──────────────────┘
          ┌──────────────┼──────────────────┐
┌──────────────┐  ┌──────────────┐  ┌──────────────┐
│ steady-shot  │▊ │   zoom-in    │▊ │ blinking-shot│▊
└──────────────┘  └──────────────┘  └──────────────┘
       │                 │                  │
┌──────────────┐  ┌──────────────┐  ┌──────────────────┐
│ keep-camera  │  │ keep-camera  │  │   keep-camera    │
├──────────────┤  ├──────────────┤  ├──────────────────┤
│ keep-viewang.│  │zoom-viewangle│  │  keep-viewang.   │
└──────────────┘  └──────────────┘  ├──────────────────┤
                                    │   blink-object   │
                                    └──────────────────┘
                                      │   │   │    │
                                   ┌────┬────┬────┬────┐
                                   │inv.│vis.│inv.│vis.│
                                   └────┴────┴────┴────┘
```

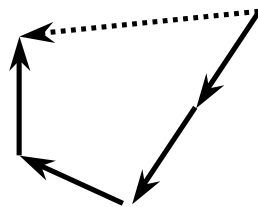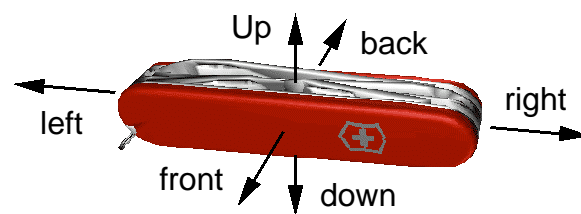| k-c	keep-camera	keep-camera
k-v	zoom-viewangle	keep-viewangle
		i | v | i | v

t ──────────▶

Animation scripts in CATHI
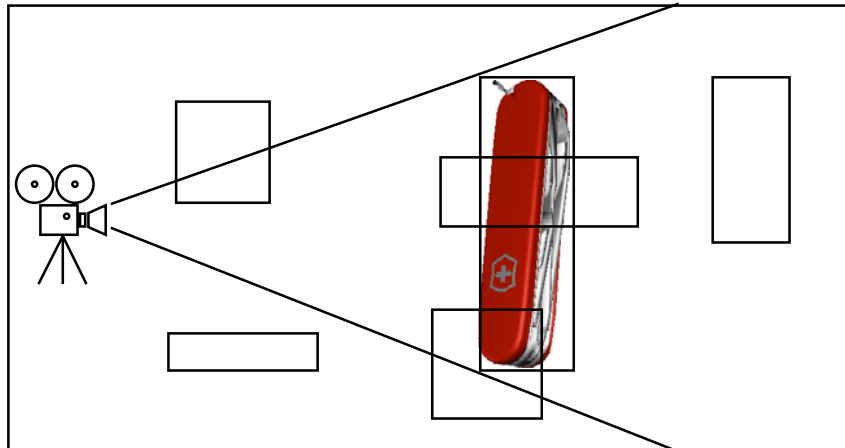
Geometrical calculations

- Calculations in the 3D model
  - Camera positions
  - Object positions and movements
  - Obstructing objects
  - Exploded views
  - Metagraphical arrows
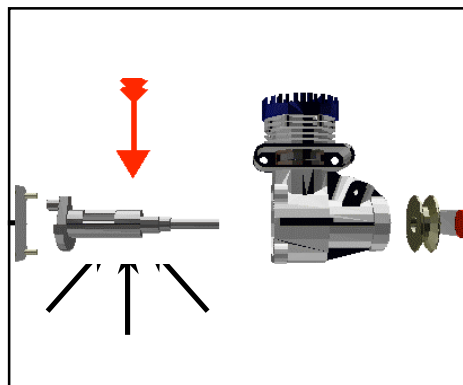
Computing camera positions

Intended viewing direction:
(front, front, left, up)

Finding obstructing objects

Positioning metagraphical arrows

Rendering

- Turn structural description into actual graphics
- Rule: no presentation without representation!
  - Structure of the output is internally represented
  - Each pixel has a "Meaning"
  - Presentation structure follows logical structure
  - User interactions can easily be interpreted
- Can be exchangeable for different output media
- Can be quite powerful
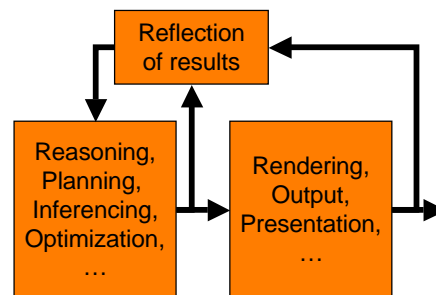 - See NPR techniques

Rendering,
Output,
Presentation,
…

Rendering in CATHI

- Translation of animation scripts into different animation languages

- Real time output to Geomview
  - Just shaded polygons
  - Ambient, distant and point lights
  - Fast rendering enables AFL

- Batch output to Renderman
  - Textures and materials
  - Spot lights
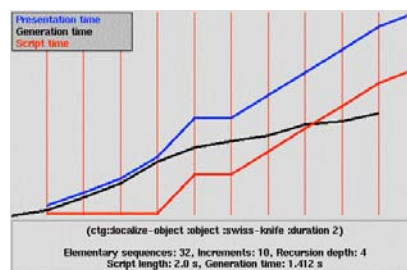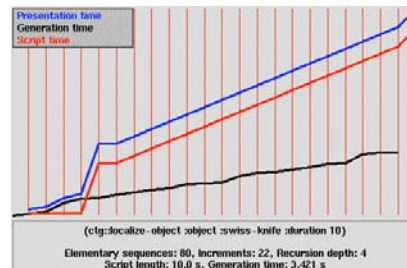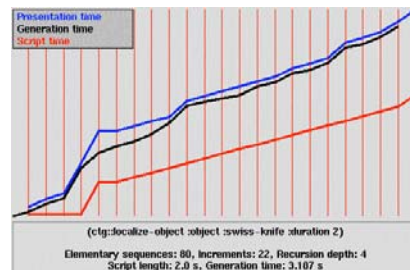 - Depth of field
  - Nice transparency
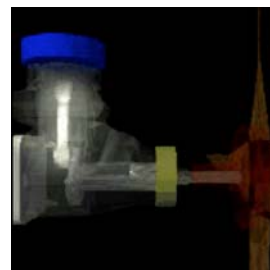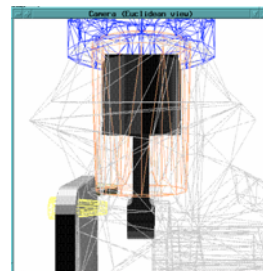
Reflection

- Analysis of the generated presentation
  - Either on the structure level
  - Or after rendering
- Influence back on the reasoning process
- Anticipation Feedback Loop (AFL)
- Can find errors in output
- Self-monitoring
- Very natural for humans
  - Bike riding
  - Speaking

Reflection of results

Reasoning, Planning, Inferencing, Optimization, …

Rendering, Output, Presentation, …

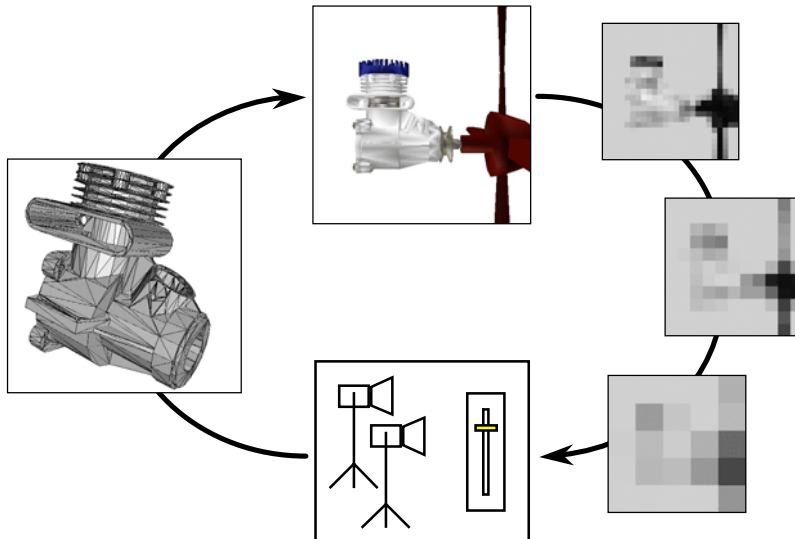# Example: Reflection on a structure level

- Temporal adaptivity of CATHI's generation process
- Choose simpler decomposition if time is scarce

Presentation time
Generation time
Script time

(ctg:localize-object :object :swiss-knife :duration 10)
Elementary sequences: 80, Increments: 22, Recursion depth: 4
Script length: 10.0 s, Generation time: 3.421 s

Presentation time
Generation time
Script time

(ctg:localize-object :object :swiss-knife :duration 2)
Elementary sequences: 80, Increments: 22, Recursion depth: 4
Script length: 2.0 s, Generation time: 3.107 s

Presentation time
Generation time
Script time

(ctg:localize-object :object :swiss-knife :duration 2)
Elementary sequences: 32, Increments: 10, Recursion depth: 4
Script length: 2.0 s, Generation time: 1.412 s

Example: Reflection after rendering
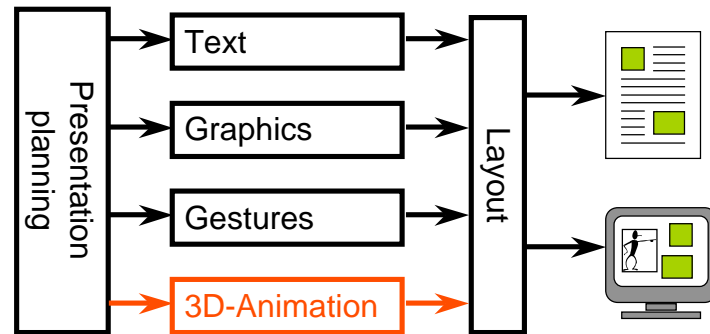
In- and Output

- Output can be just graphical or coordinated with other media
  - Coordination by „higher authority"
  - Integration of other media in the planning process
- Input can be explicit or implicit
  - Checking boxes, setting user profile
  - Previous interactions with the system
  - Learned profile
- In CATHI: just checkboxes

Display

input

Integration of CATHI into WIP

Some example generations of CATHI

Adaptation to different capabilities of the output medium