Smart Graphics: Methoden 1

Vorlesung „Smart Graphics”

Andreas Butz, Otmar Hilliges

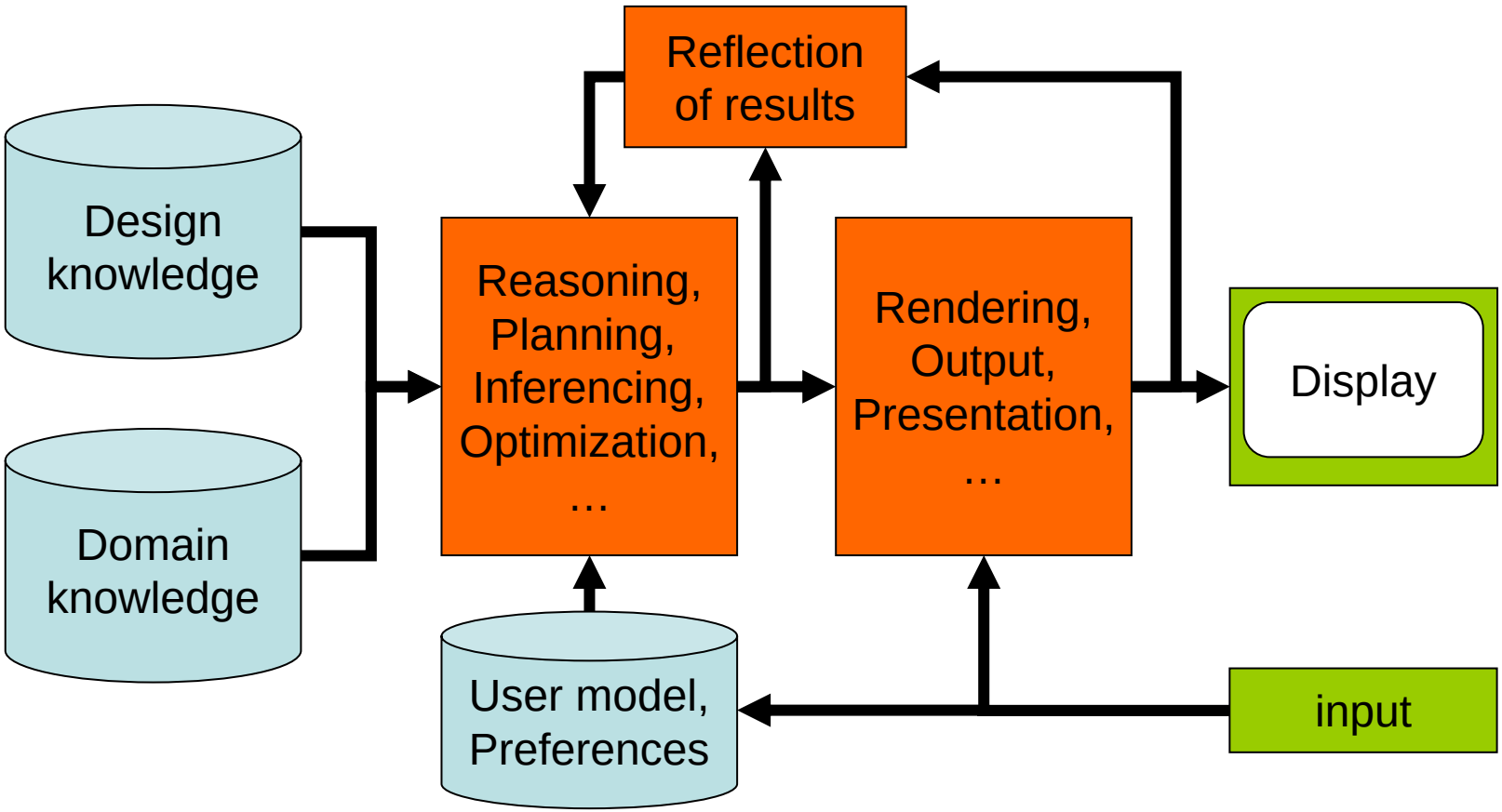
Mittwoch, 16. November 2005

Themen heute

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

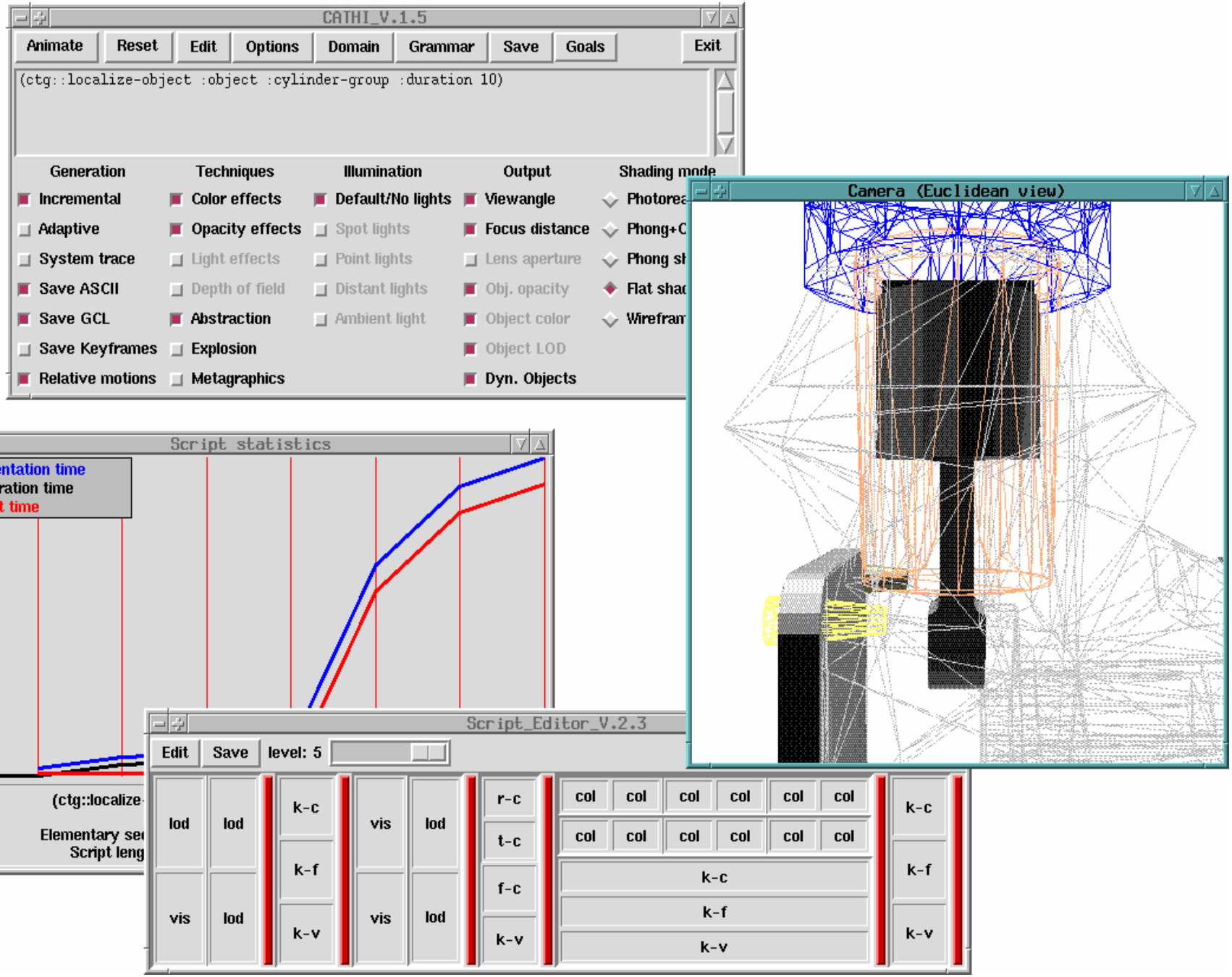
Some typical elements of SG systems

- Strong simplification and generalization
- Often only some elements present



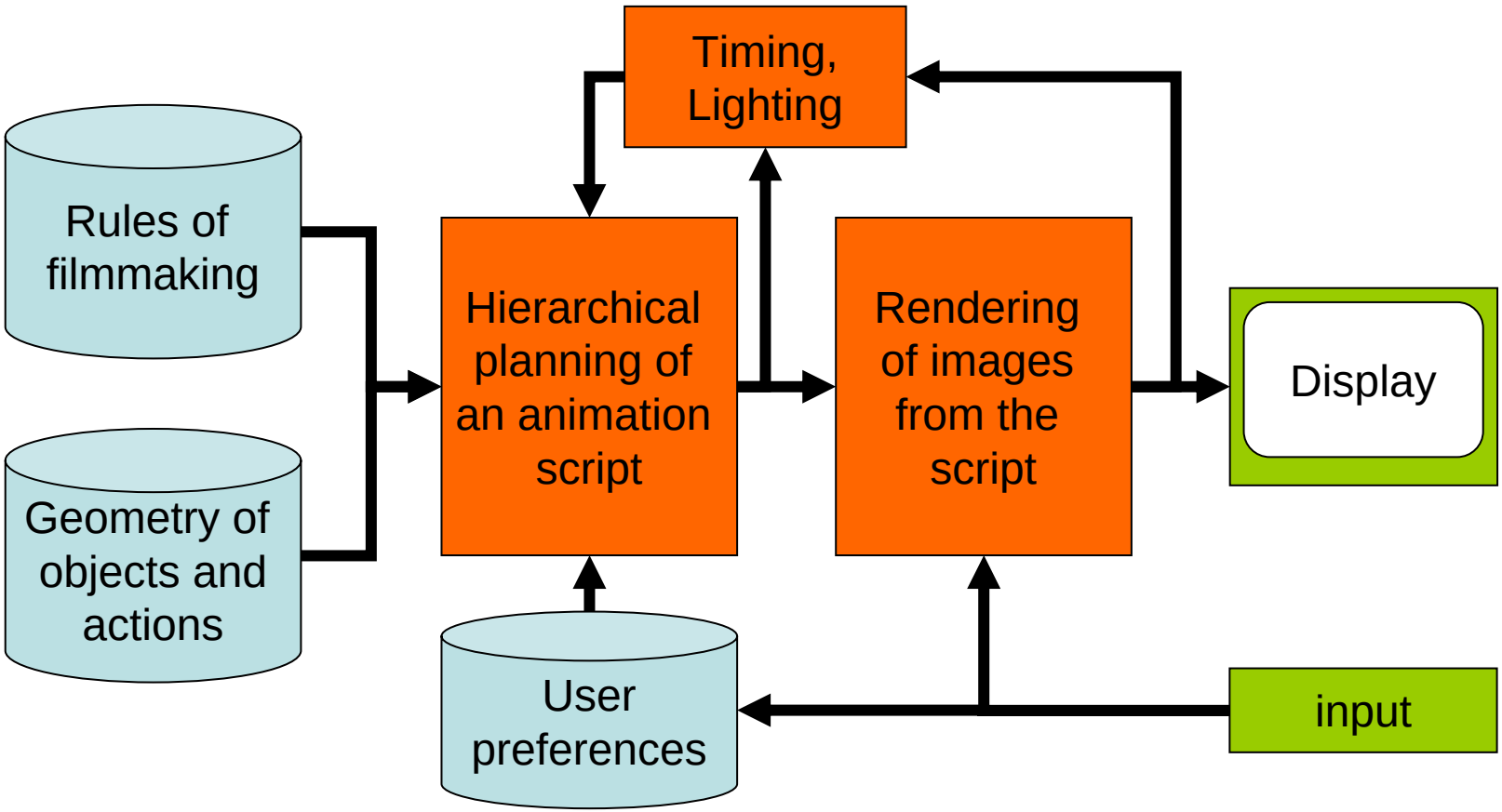
Concrete example: filmmaking

- 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


CATHI_V.1.5
Animate
Reset
Edit
Options
Domain
Grammar
Save
Goals
Exit
(ctg::localize-object :object :cylinder-group :duration 10)
Generation
Incremental
Adaptive
System trace
Save ASCII
Save GCL
Save Keyframes
Relative motions
Techniques
Color effects
Opacity effects
Light effects
Depth of field
Abstraction
Explosion
Metagraphics
Illumination
Default/No lights
Spot lights
Point lights
Distant lights
Ambient light
Output
Viewangle
Focus distance
Lens aperture
Obj. opacity
Object color
Object LOD
Dyn. Objects
Shading mode
Camera (Euclidean view)
Script statistics
Presentation time
Generation time
Script time
Script_Editor_V.2.3
Edit
Save
level: 5
lod
vis
k-c
k-f
k-v
r-c
t-c
f-c
col
k-c
k-f
k-v

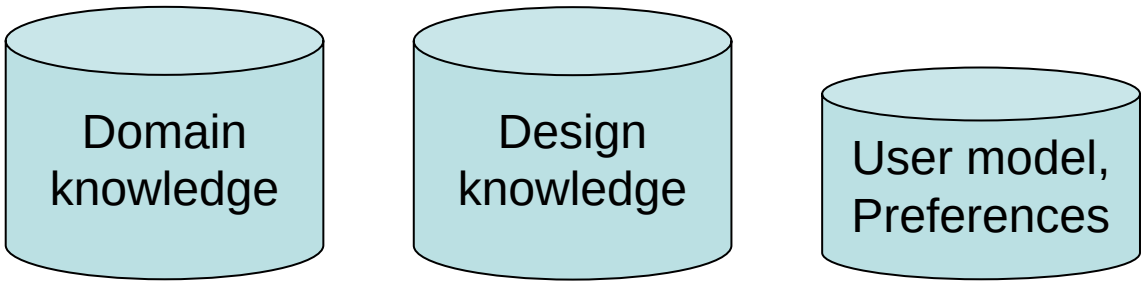
Concrete example: filmmaking

- Example system CATHI [Butz, 97]



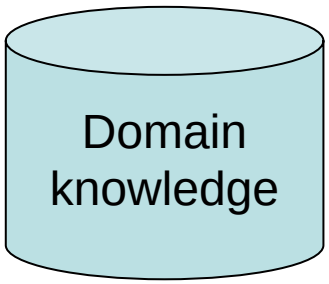
Knowledge representation

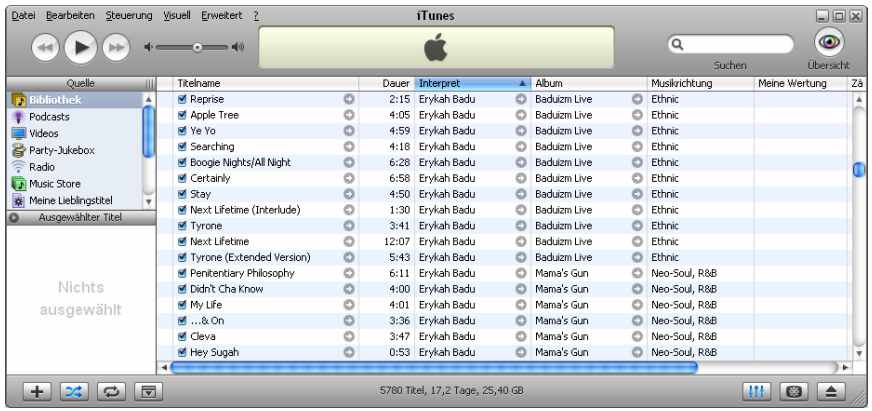
- Representations can only capture part of the reality
  - Which aspects do we need to model?
  - At which level of detail do we need to model them?
  - Do we need qualitative or quantitative knowledge?
  - How do we want to process the knowledge?
- Different kinds of knowledge must be represented
 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

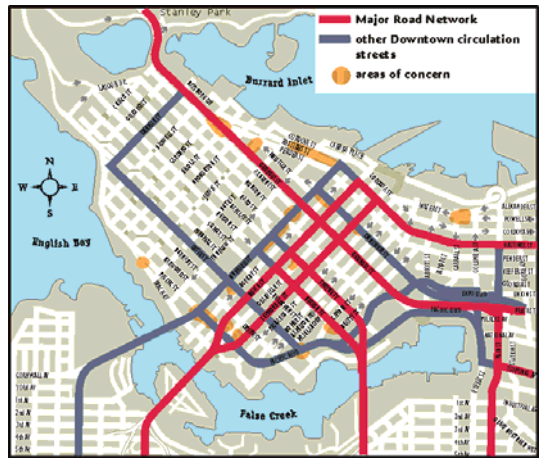


Domain Knowledge

- 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

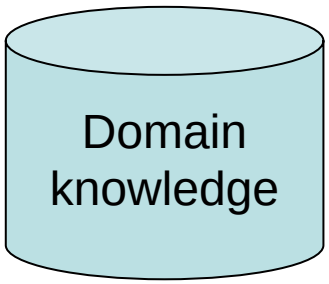


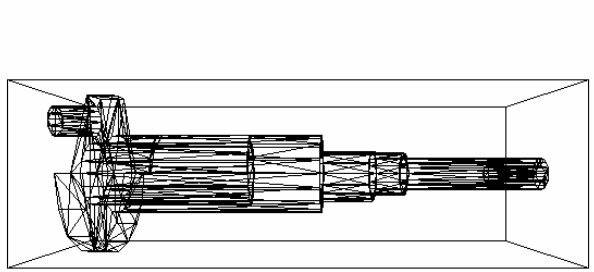


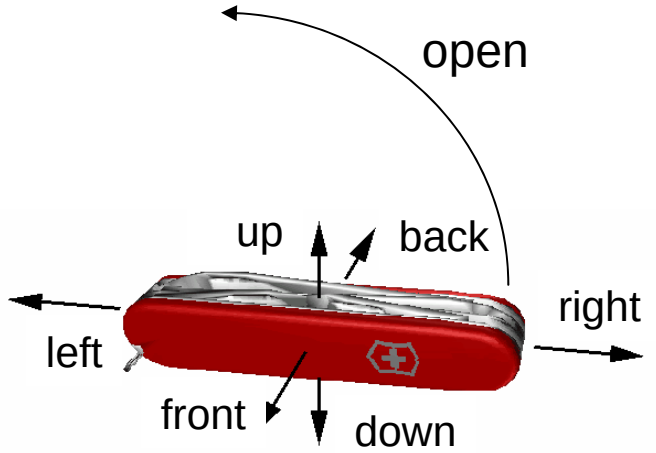


Domain Knowledge

- 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

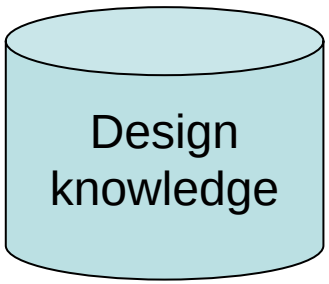


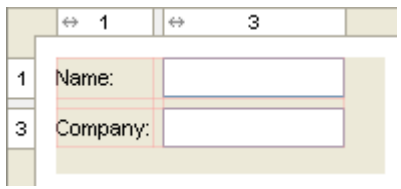


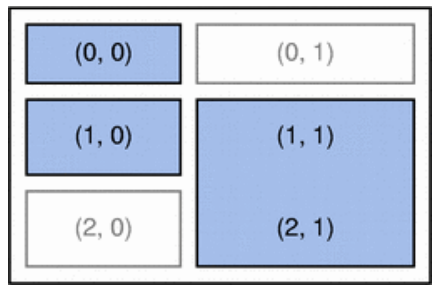


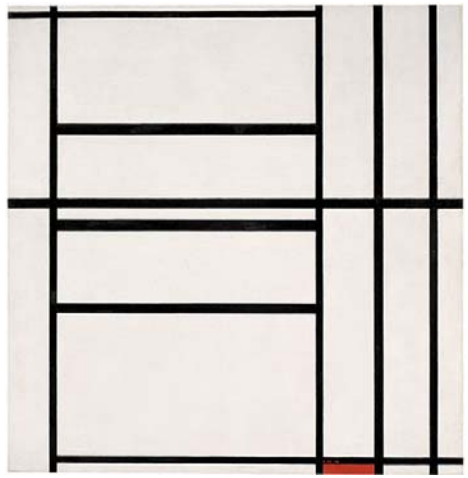
Design Knowledge

- 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



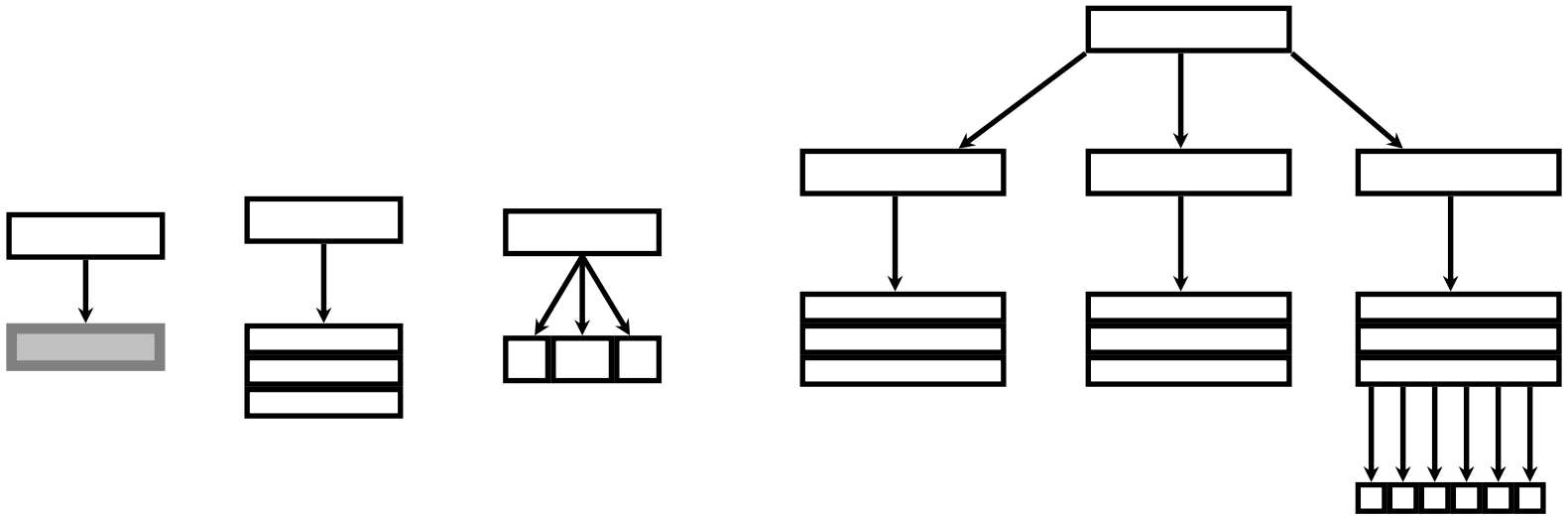






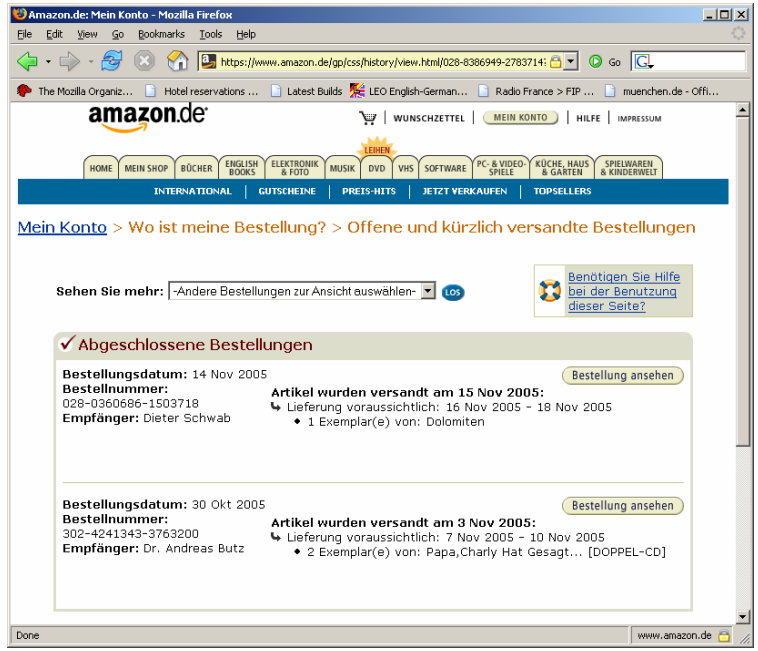
Example from CATHI: a formal grammar of the film language

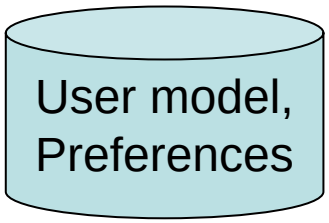
- Rules for decomposing sequences into subsequences
- Reusable in different situations
- Querying calculations in the 3D model
- Details later



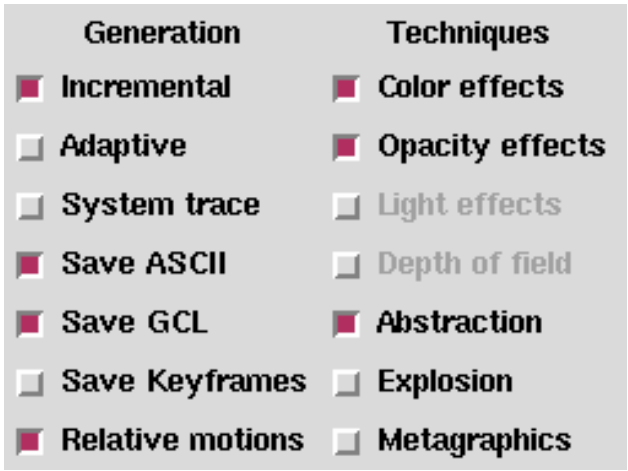
User model / preferences

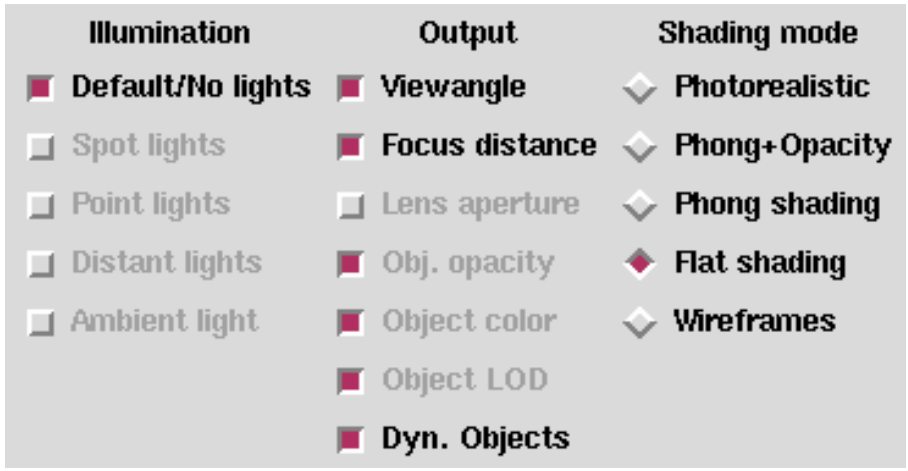
- Knowledge about the user
  - Properties, such as level of expertise
  - Preferences, given implicitly or explicitly
  - Current context of the user
  - Also: capabilities of the output medium
- Examples
  - Previously bought items
  - Personal viewing preferences
  - Current resolution of the output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





User preferences in CATHI





Stylistic preferences of the user

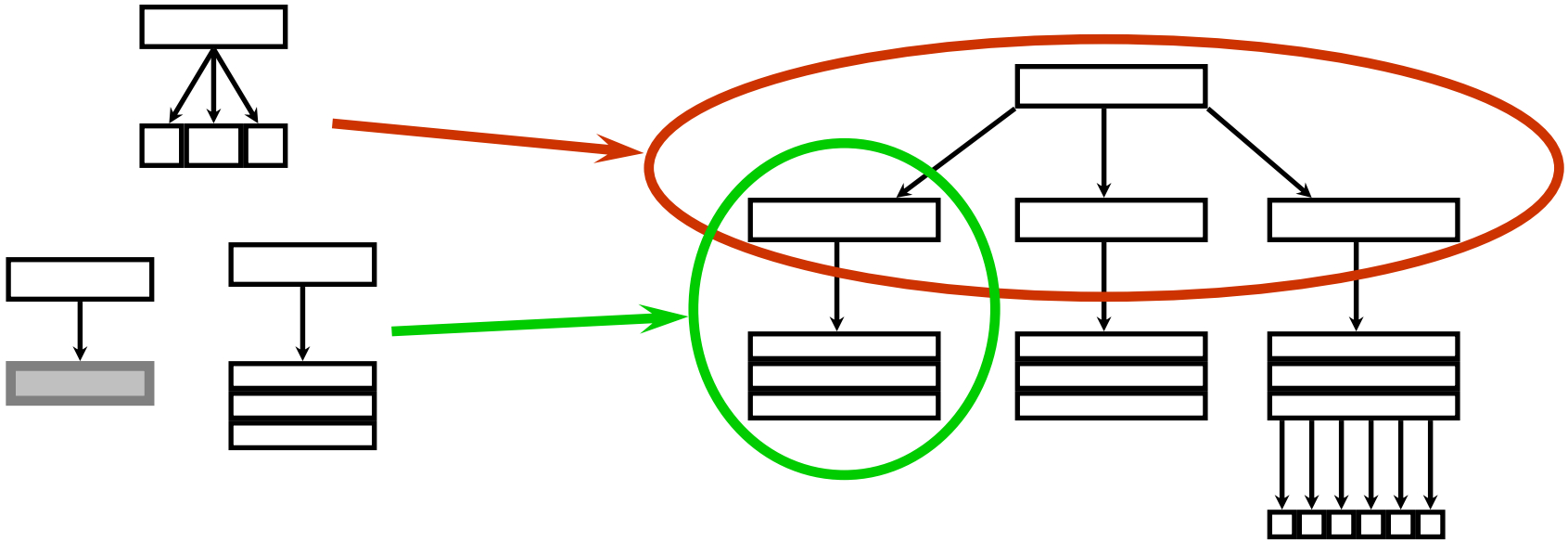
Graphical capabilities of the user‘s machine (back in 1997!)

Reasoning

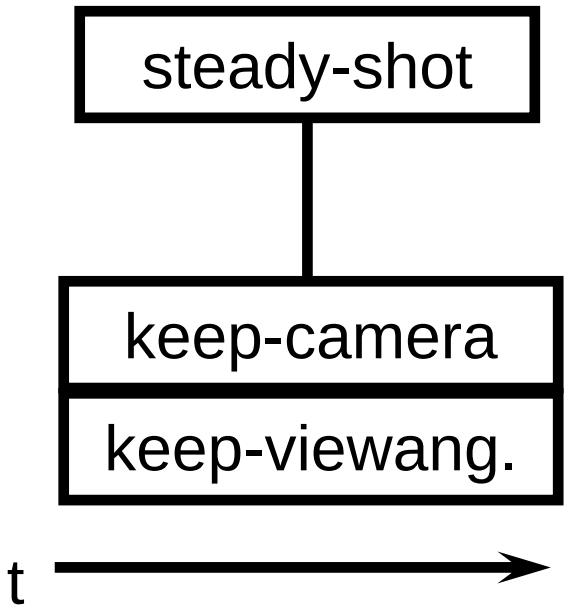
- Algorithms for:
  - Further refining the domain knowledge
  - Application of design knowledge
- Output of the reasoning process:
  - Complete structural description of the presentation
- Examples:
  - Route calculation on a road network
  - Layout of labels in a map
  - Layout of text blocks on a page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

Reasoning,
Planning,
Inferencing,
Optimization,
…

Example: animation scripts

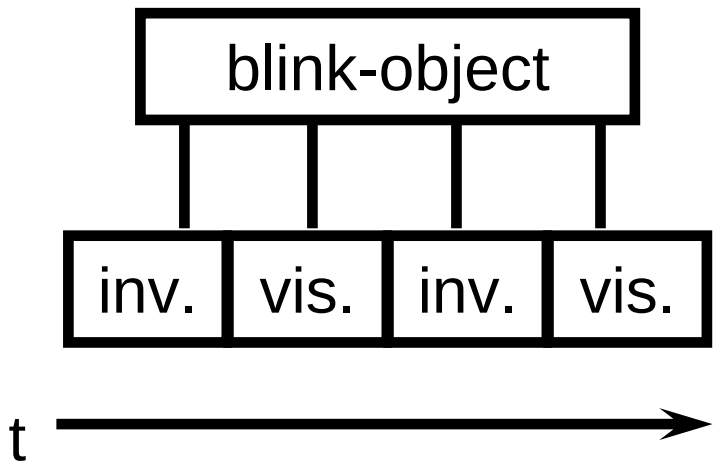


Parallel Decomposition




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

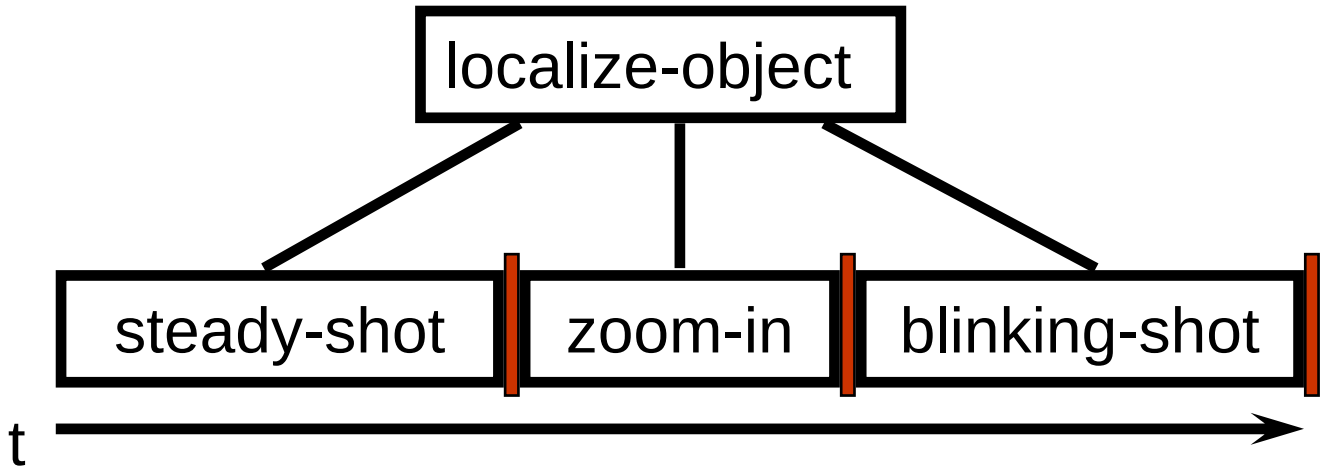
Sequential Decomposition



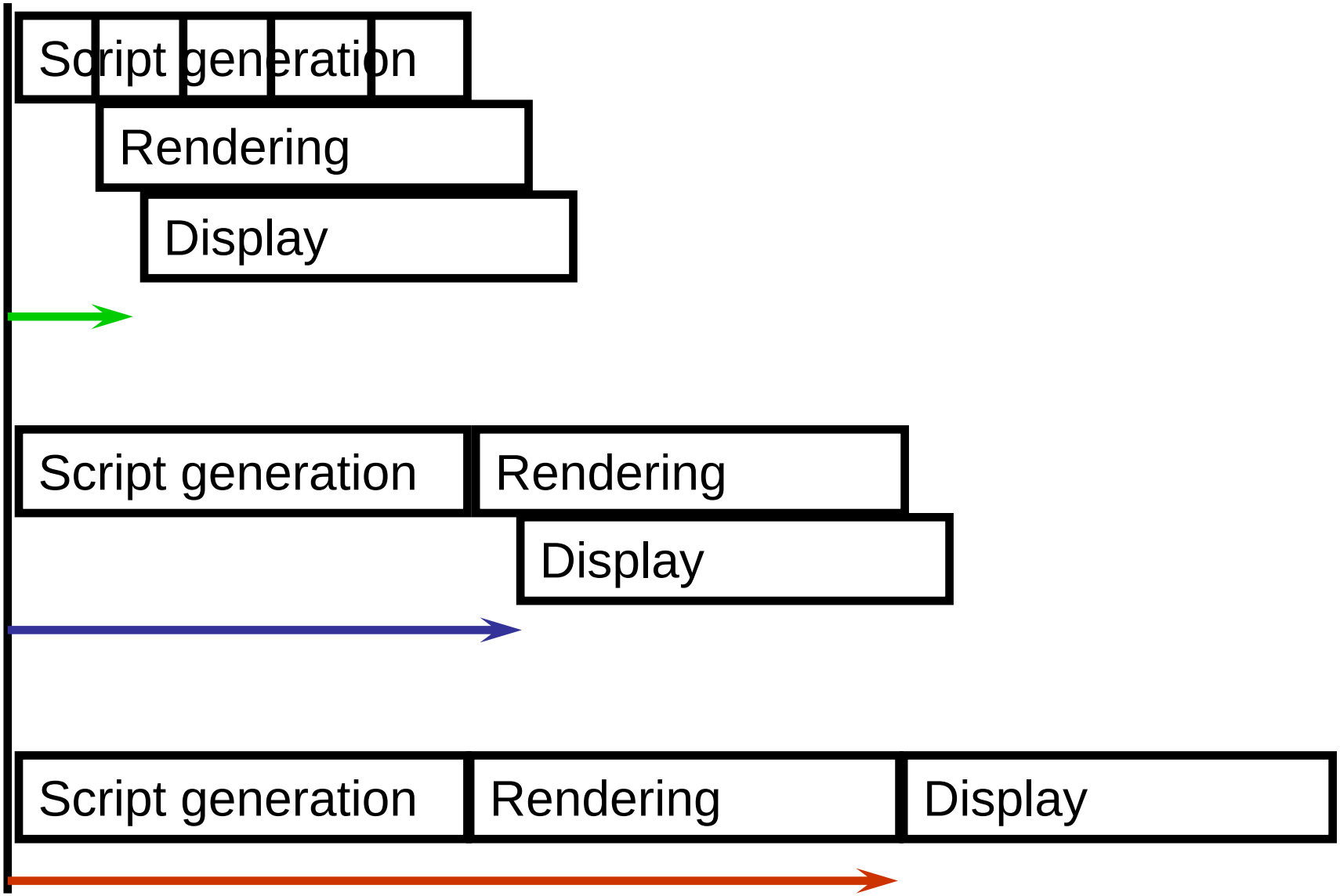

```
(defrule blink-object (object duration)
  (sequential
    (invisible object (* 0.25 duration))
    (visible object (* 0.25 duration))
    (invisible object (* 0.25 duration))
    (visible object (* 0.25 duration))))
```

Incremental Decomposition

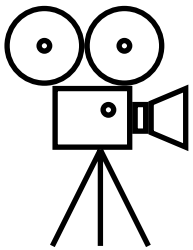
```
(defrule localize-object (object duration)
  (incremental
    (steady-shot (* 0.2 duration))
    (zoom-in object (* 0.4 duration))
    (blinking-shot object (* 0.4 duration))))
```



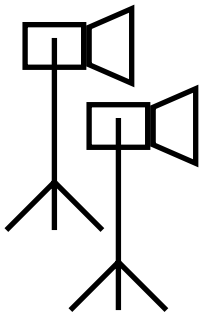
Why incremental generation?



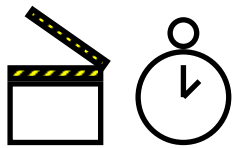
Current generation context



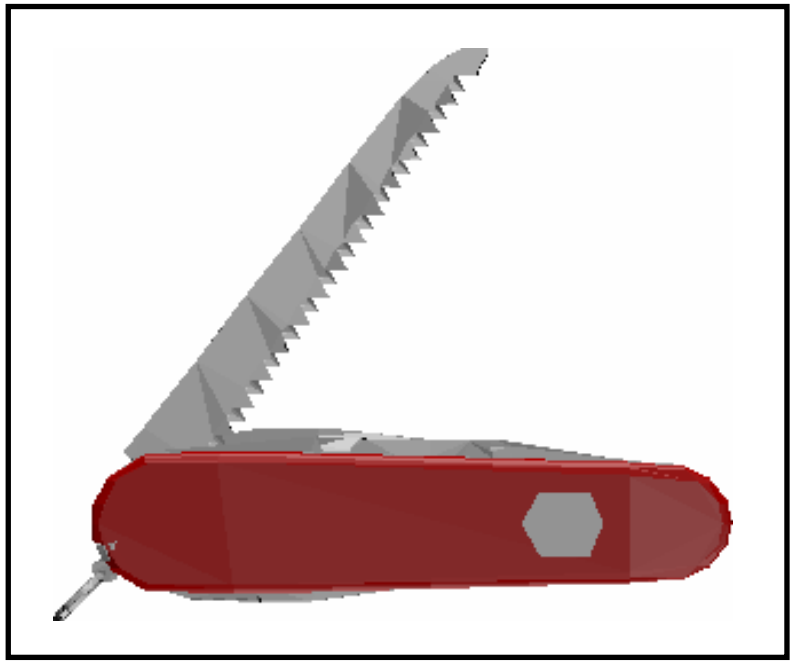
Camera position and settings



Base lighting
Effect lights



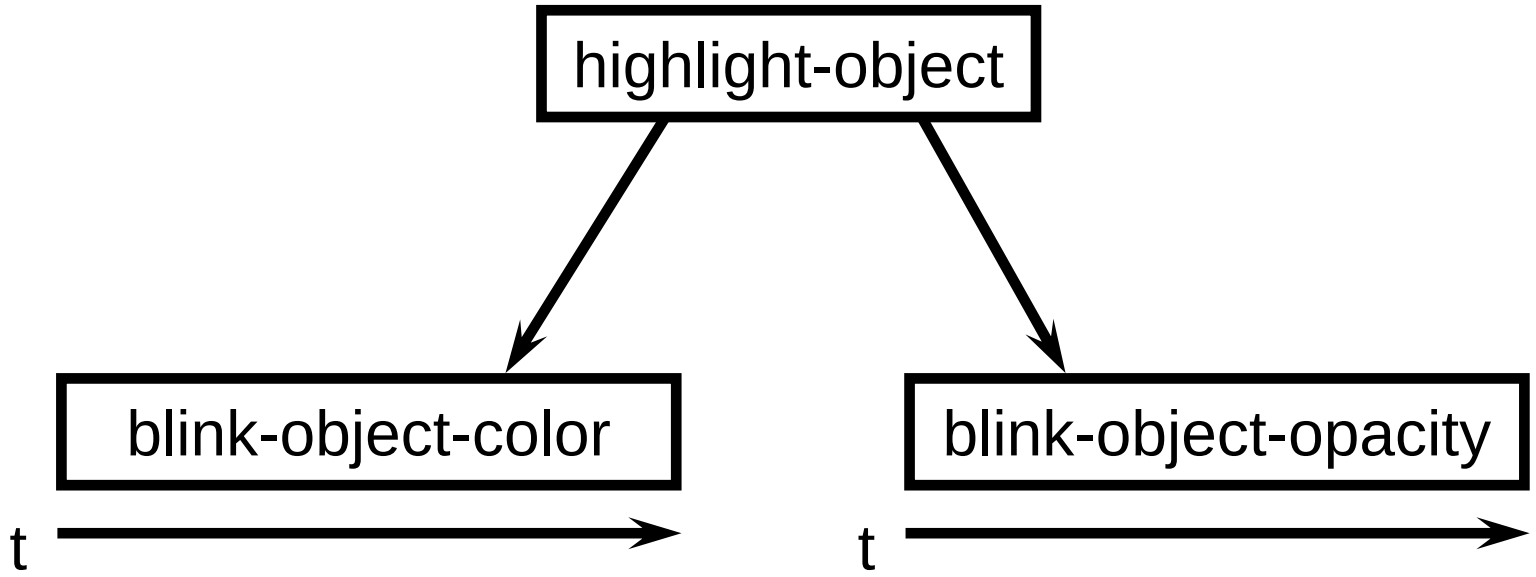
Timing of the generation and presentation



Object positions and properties

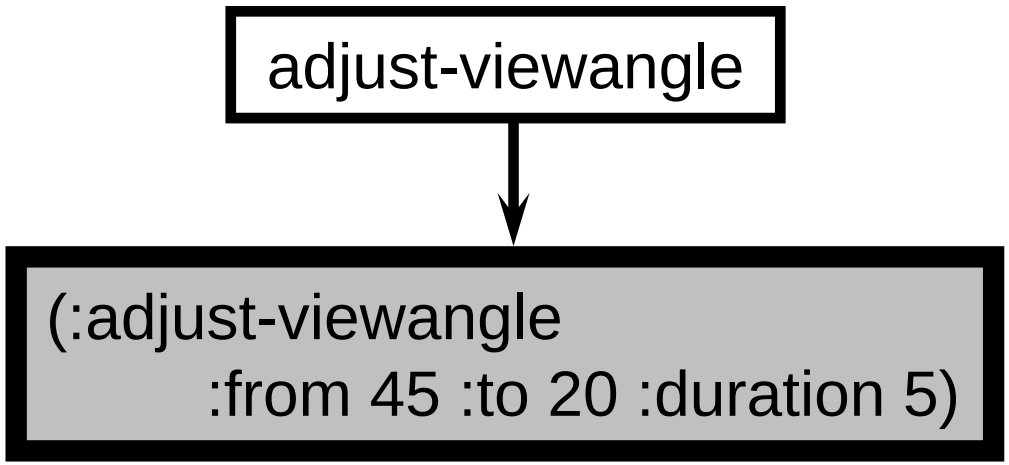
Conditional Decomposition

```
(defrule highlight-object (object duration)
  (if (feature color)
      (blink-object-color object duration)
    (blink-object-opacity object duration)))
```

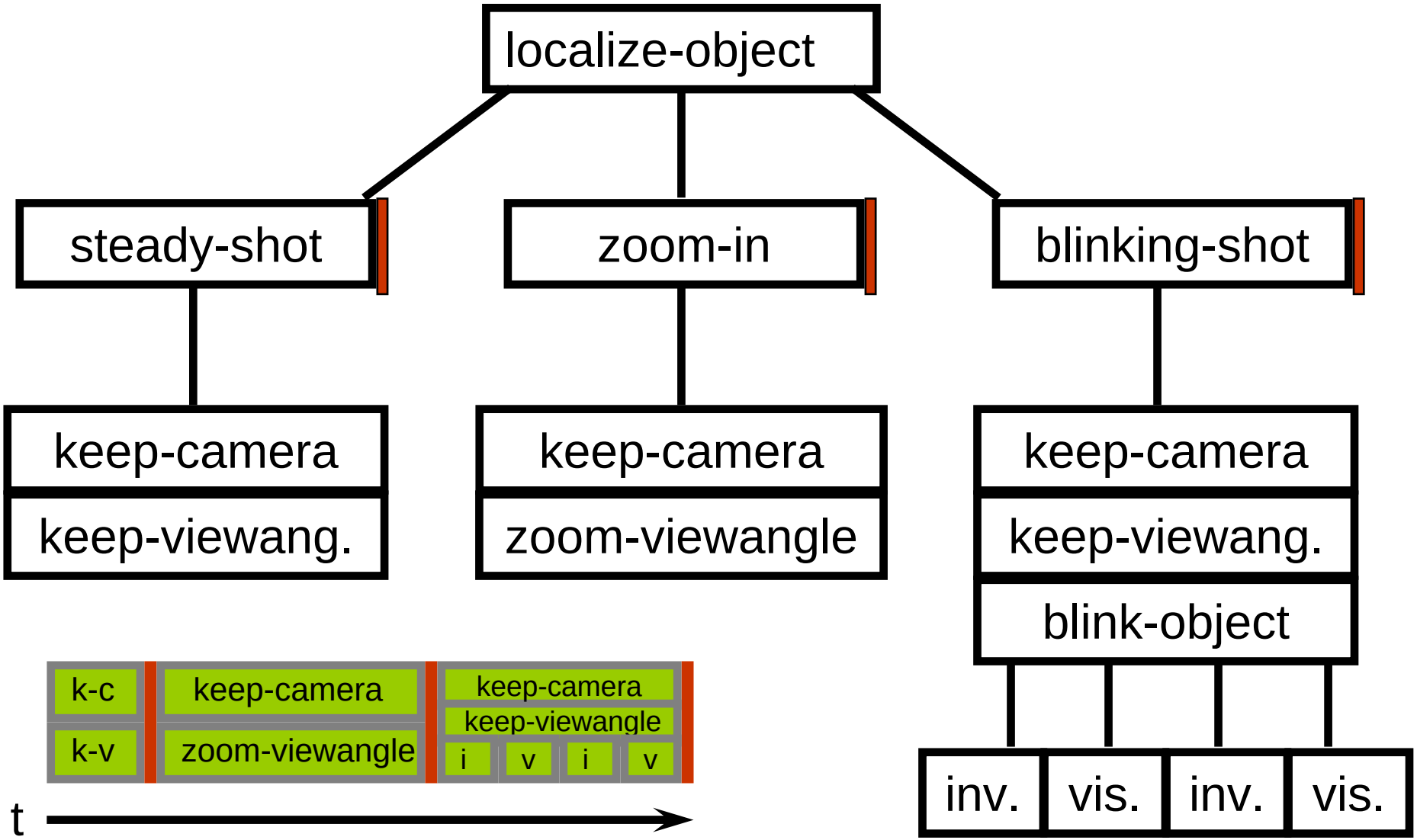


Translation of elementary sequences

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



Generierung eines Skripts

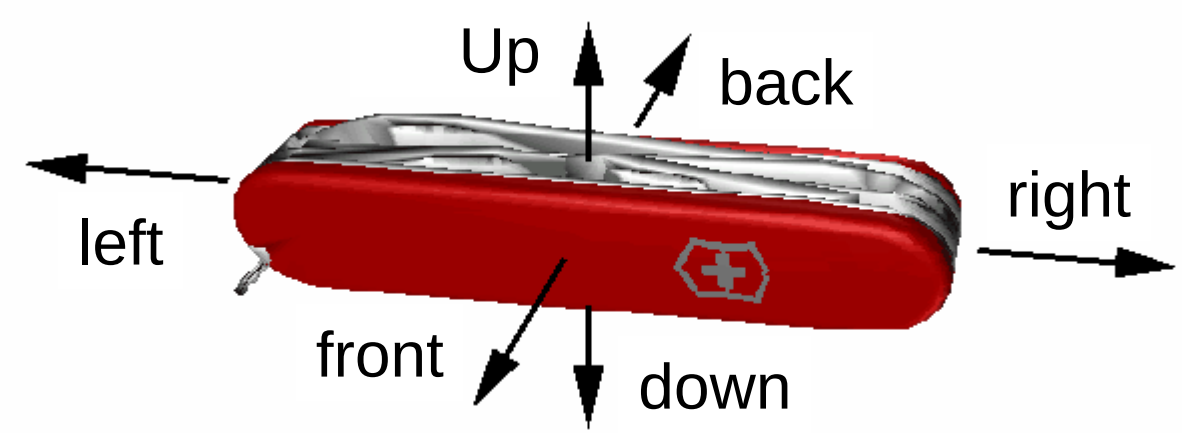


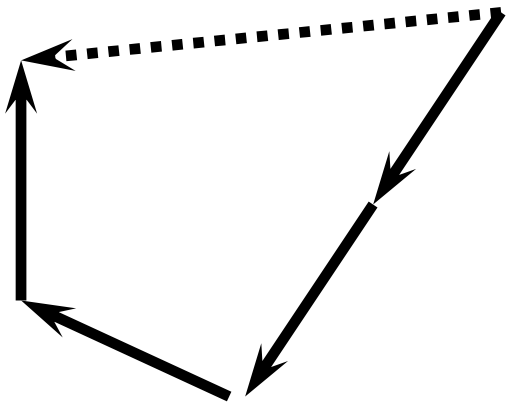
Animation scripts in CATHI

Geometrical calculations

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

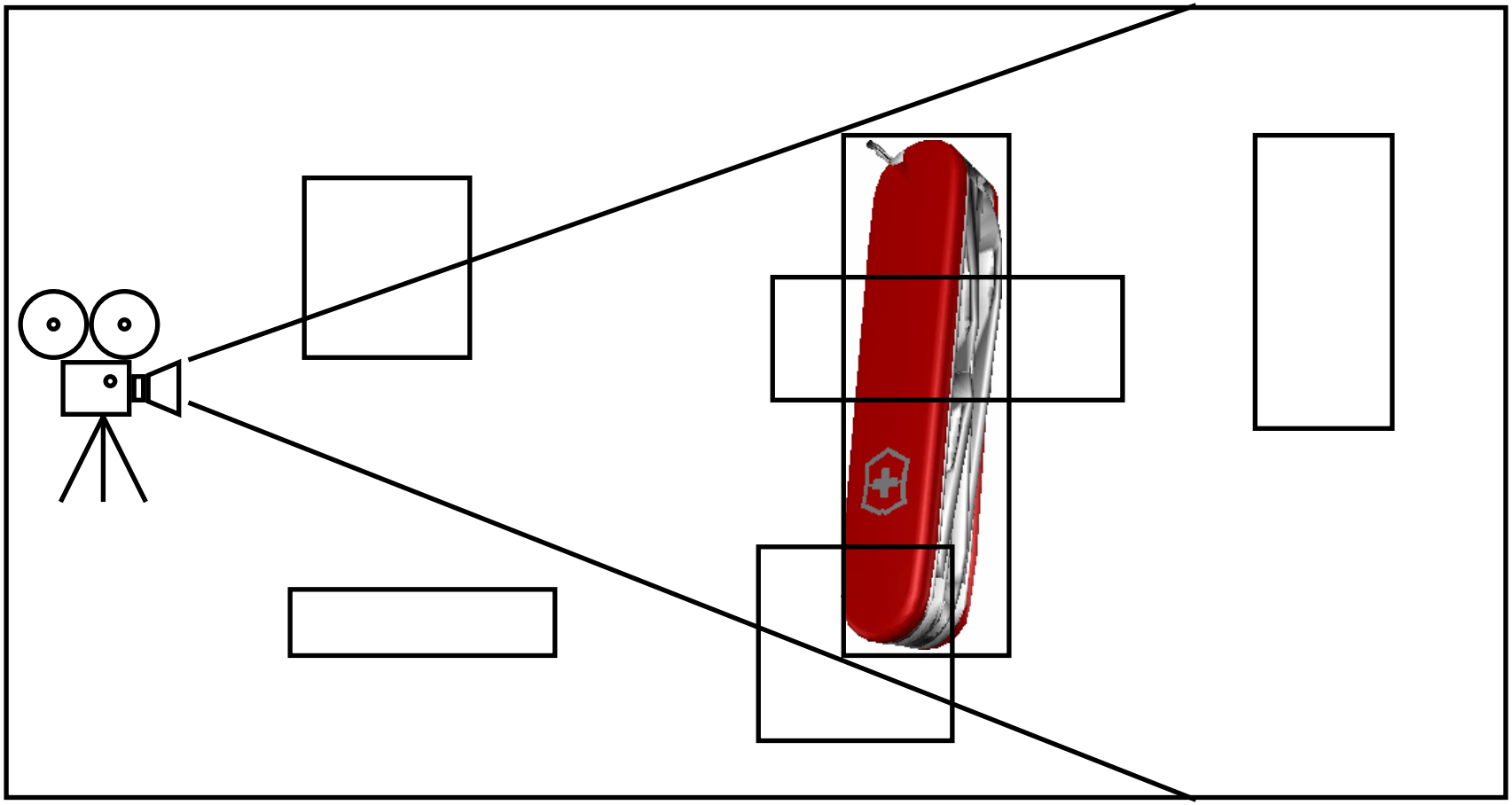
Computing camera positions



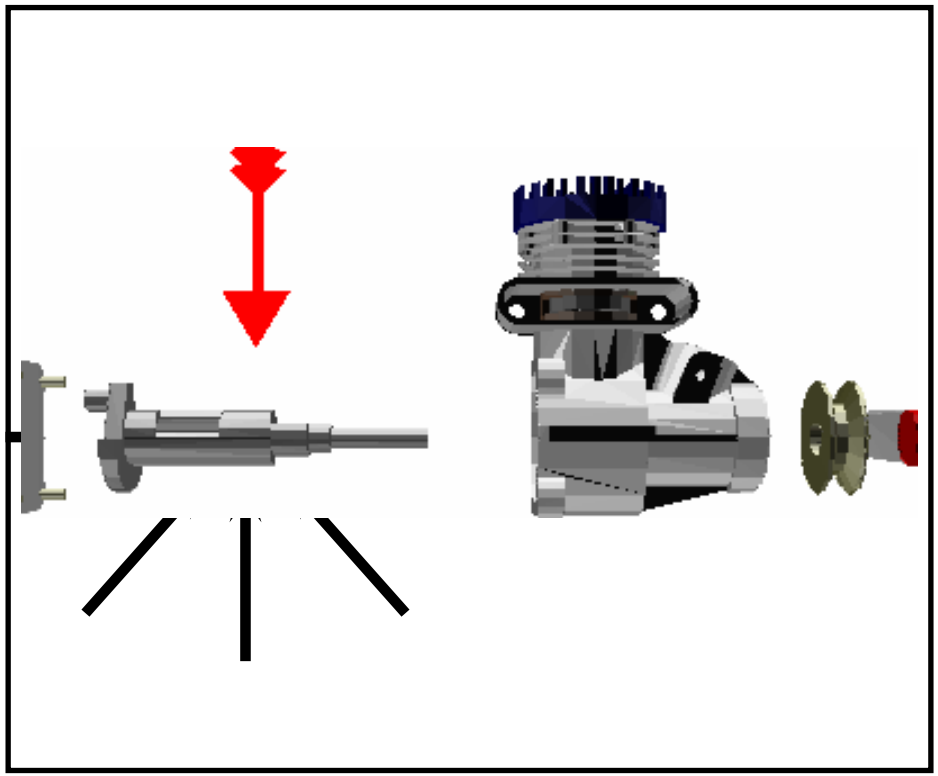


Intended viewing direction:
(front, front, left, up)

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

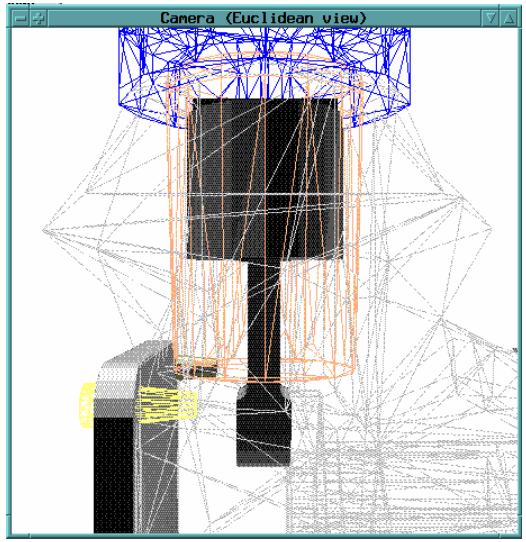
Rendering, Output, Presentation, …

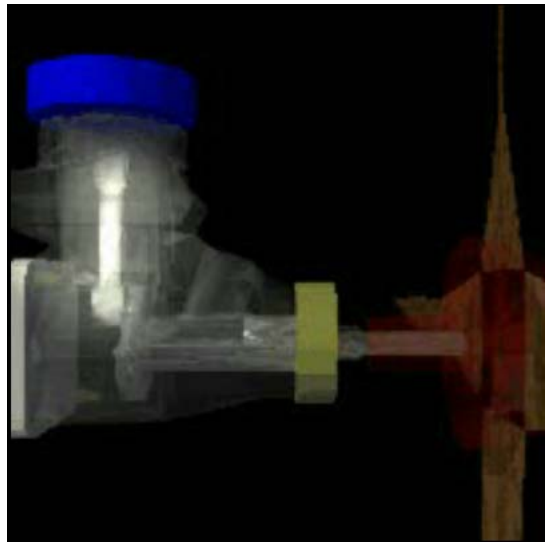
Rendering in CATHI

- 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

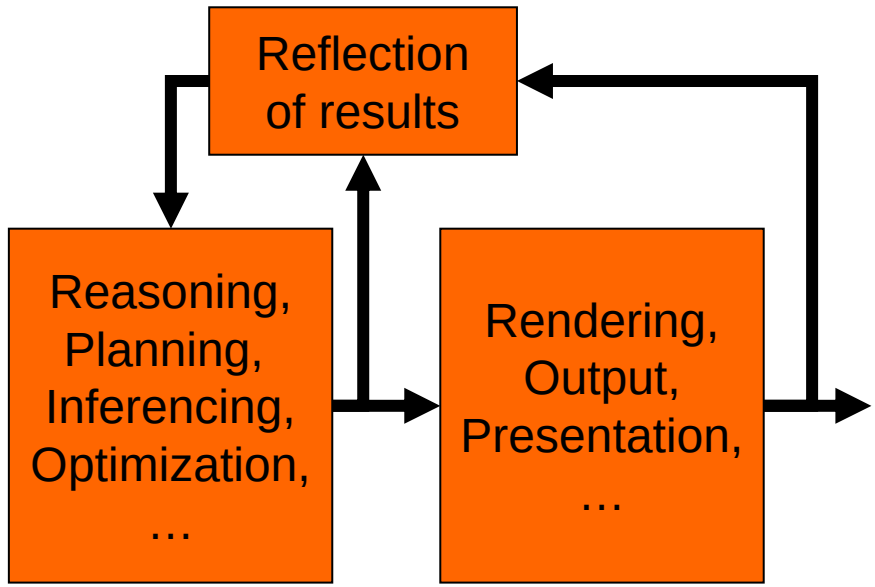
- Batch output to Renderman
  - Textures and materials
  - Spot lights
  - Depth of field
  - Nice transparency





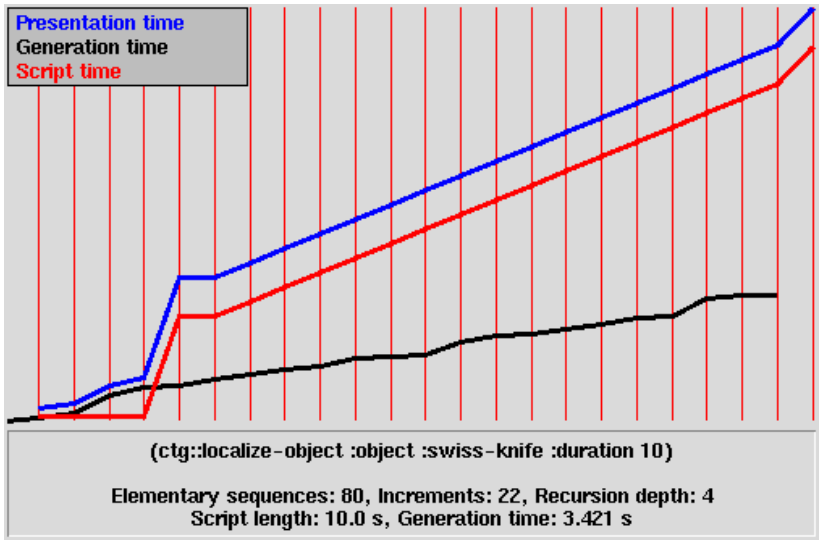
Reflection

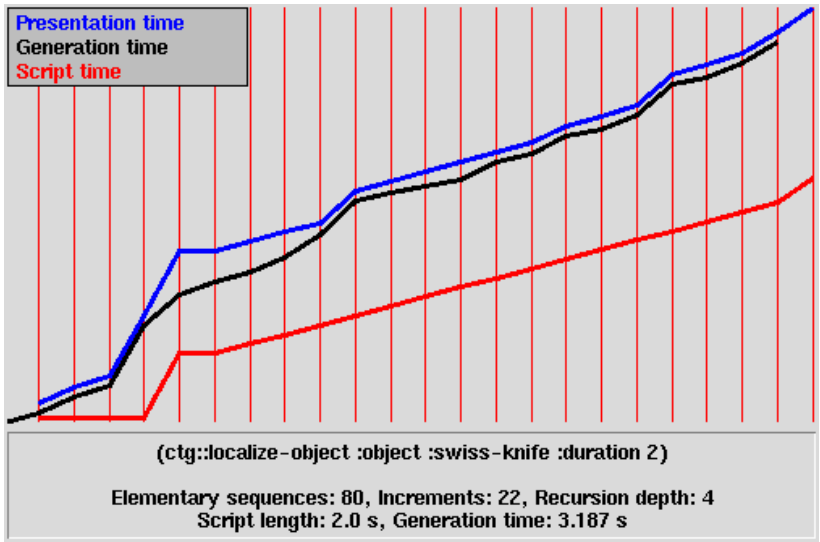
- Analysis of the generated presentation
  - Either on the structure level
  - Or after rendering
- Influence back on the reasoning process
- Anticipation Feedback Loop (AFL)
- Can find errors in output
- Self-monitoring
- Very natural for humans
  - Bike riding
  - Speaking

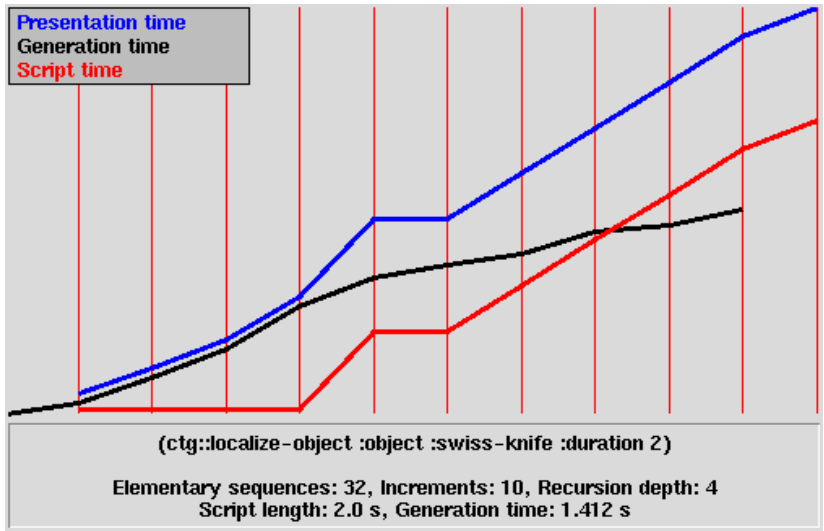


Example: Reflection on a structure level

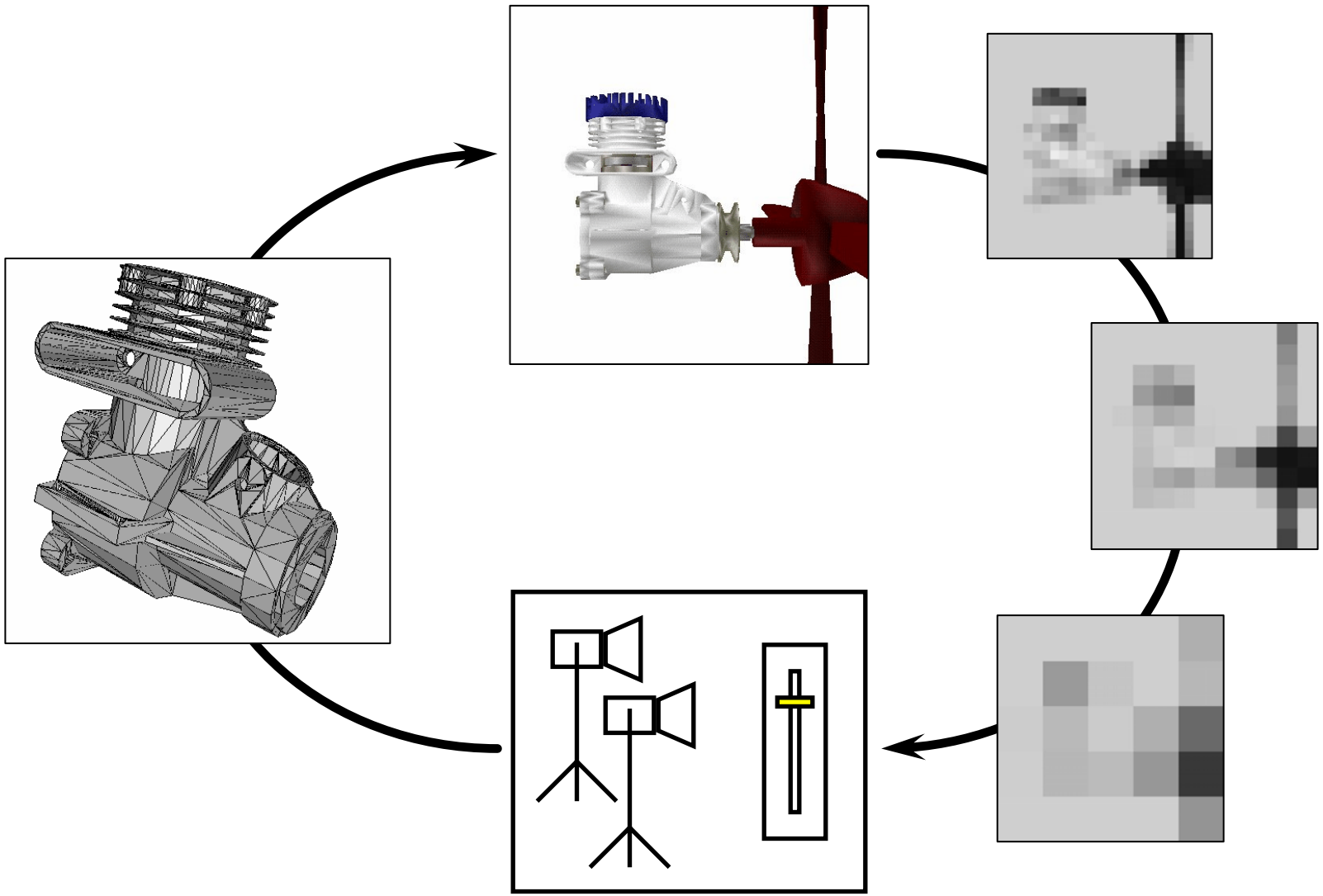
- Temporal adaptivity of CATHI‘s generation process
- Choose simpler decomposition if time is scarce





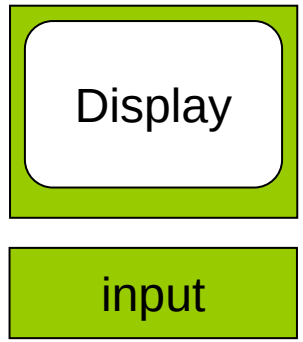


Example: Reflection after rendering

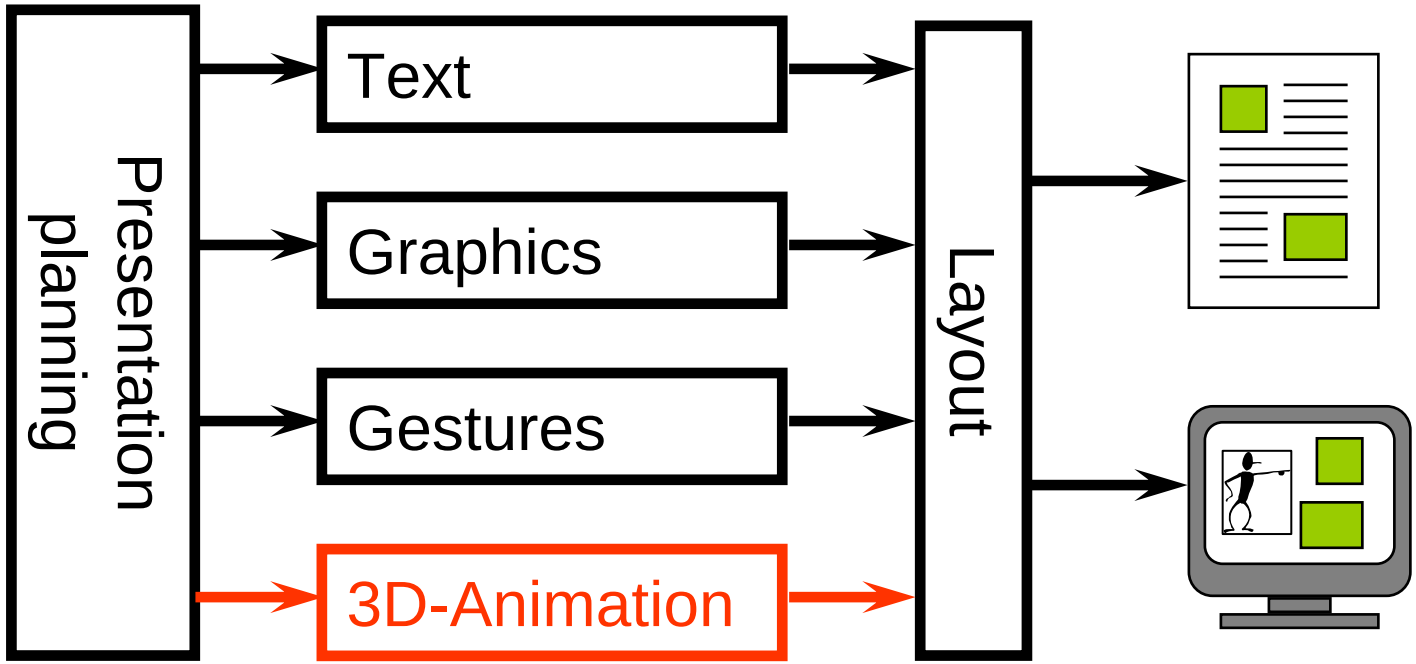


In- and Output

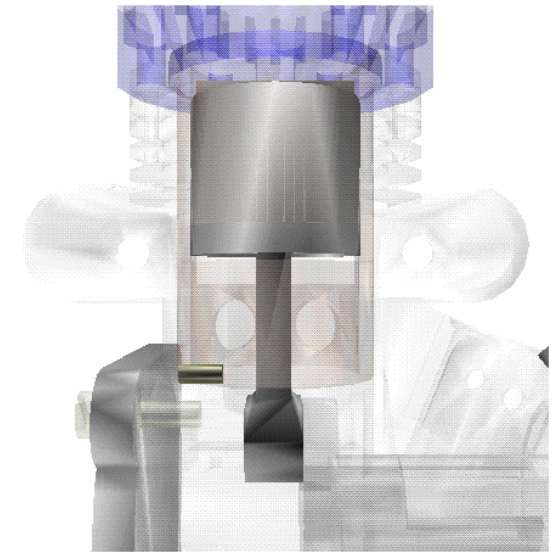
- Output can be just graphical or coordinated with other media
  - Coordination by „higher authority“
  - Integration of other media in the planning process
- Input can be explicit or implicit
  - Checking boxes, setting user profile
  - Previous interactions with the system
  - Learned profile
- In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

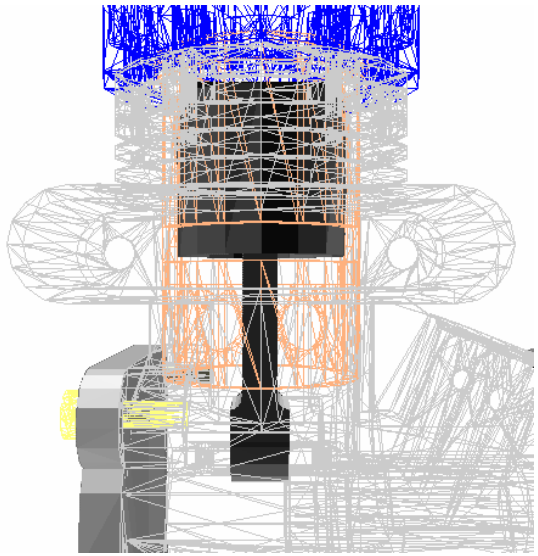


Integration of CATHI into WIP



Some example generations of CATHI





Adaptation to different capabilities of the output medium