UNIKASSEL
VERSITÄT

SOFTWARE ENGINEERING
UNIVERSITÄT KASSEL

Software Tool Construction

Wintersemester 2023/24

Adrian Kunz

Vorlesung 13

Programmiersprachen: Uneindeutigkeit

- `(a) * b`
  - a Variable: "Berechne a mal b"
  - a Typname: "Caste den Werte von Pointeradresse b zu a"
- `T(*b)[4]`
  - T Funktion: "Rufe Funktion T auf mit dem Wert an Pointeradresse b und nimm Element an Index 4
  - T Typname: "Deklariere b als Pointer zu einem Array von 4 T's"

- `if(a) if (b) x++; else y++;`
  -
```
if (a)
  if (b)
    x++;
else
  y++;
```
  -
```
if (a)
  if (b)
    x++;
  else
    y++;
```

- https://stackoverflow.com/questions/59482460/how-to-handle-ambiguity-in-syntax-like-in-c-in-a-parsing-expression-grammar
- https://langdev.stackexchange.com/questions/9/why-do-some-programming-languages-choose-to-have-a-dedicated-keyword-for-elseif

DSLs für Domänenexperten

$$\text{coord} = \begin{bmatrix} 2 \\ 3 \\ 1 \end{bmatrix}$$

$\theta = 45°$

$$\text{rotation} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\text{rot\_coord} = coord \cdot rotation$$

```
coord = [2, 3, 1]

theta = np.radians(45)

rotation = np.array([
  [np.cos(theta), -np.sin(theta), 0],
  [np.sin(theta), np.cos(theta), 0],
  [0, 0, 1]
])

rot_coord = np.dot(
  rotation, np.array(coord)
)
```

DSLs: Unicode

$$\text{coord} = \begin{bmatrix} 2 \\ 3 \\ 1 \end{bmatrix}$$

$$\theta = 45°$$

$$\text{rotation} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\text{rot\_coord} = coord \cdot rotation$$

```
coord = [ 2 ]
        [ 3 ]
        [ 1 ]

θ = 45°

rotation = [ cos(θ) -sin(θ) 0 ]
           [ sin(θ)  cos(θ) 0 ]
           [   0       0    1 ]

rot_coord = coord · rotation
// Unicode: [ │ ⌊ ⌋ ⎢ ⎥ θ ·
```

MPS

Meta-Programming System von Jetbrains

MPS: DSLs

```
System.out.println(String.valueOf((∑ [[1 k 0] [0 1.0 0] [0 0 1]])));

System.out.println(exp(a + i * b) - exp(a) * (cos(b) + i * sin(b)));

matrix<Double> s = [[3.0] [sin(1)] [1] [1]
                    [3^2] [1] [3 + 1.0/2] [2]
                    [0] [7 - 1.0/2 + 1] [exp(1)] [3]
                    [4] [2] [0] [0]];
```

- https://www.jetbrains.com/mps/

MPS: Projectional Editor

DecisionSample

```
  Money discount;
  discount = create(createPerson());

  if (discount > 400 USD || discount >= 350 EUR) {
    discount = 300 EUR;
  }

  System.out.println("Your name: " + createPerson());
  System.out.println("Your discount: " + discount);

}

public Money create(map<string, Object> person) {
  return Money Default: 0 EUR
```

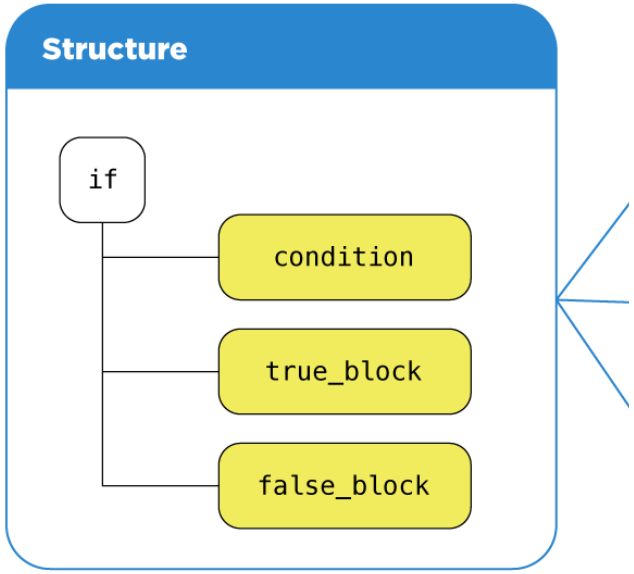
	isLevel_1(person)	*isLevel_2*(person)
isChild(person)	500 EUR	1000 EUR
isAdult(person)	50 EUR + this.seasonalBonus()	100 EUR + this.seasonalBonus()
isRetired(person)	200 EUR	250 EUR + (person["name"] == "Susan" ? this.seasonalBonus() : 0 EUR)

```
;
}

private Money seasonalBonus() {
  return 100 EUR;
}
```

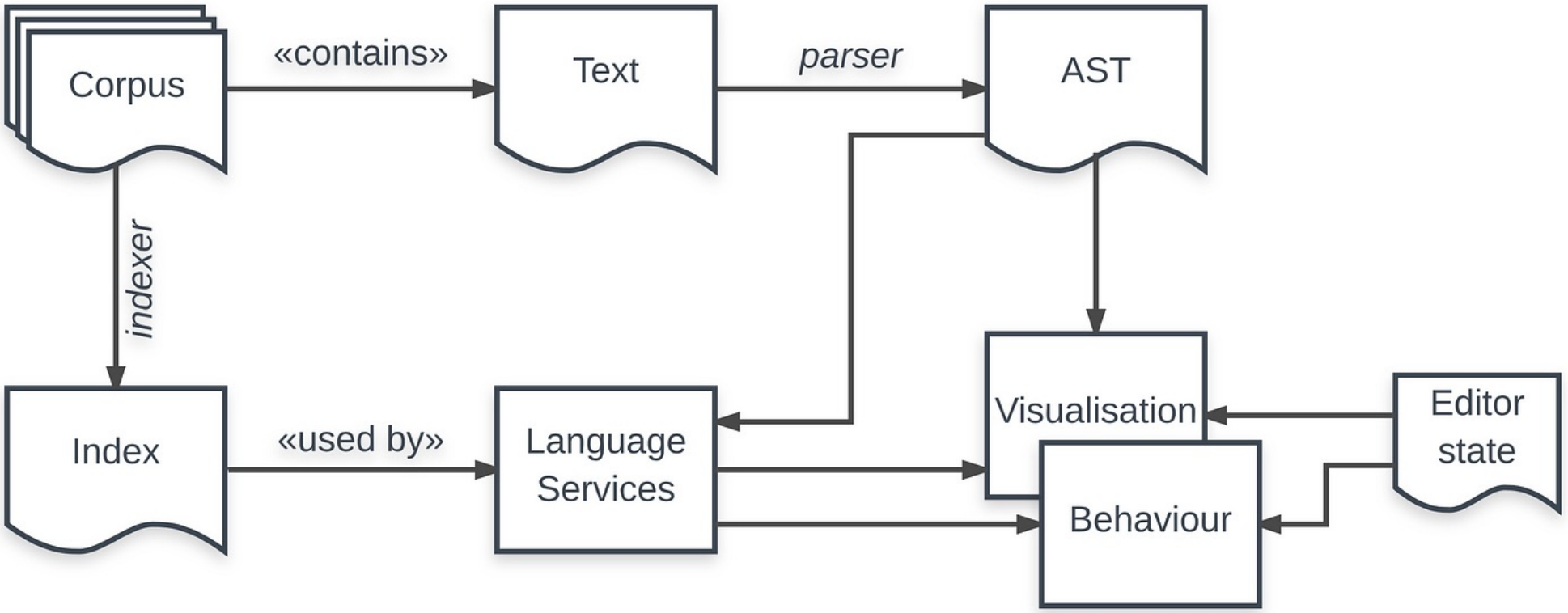
- https://www.jetbrains.com/mps/

Structural + Projectional Editing



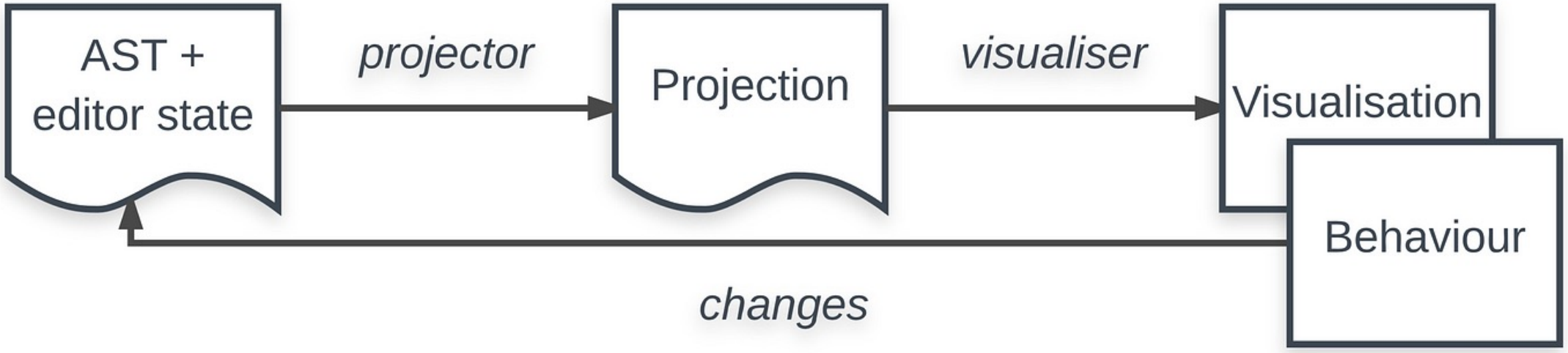
- https://www.jetbrains.com/help/mps/basic-notions.html#projectionaleditor

Textueller Editor: Architektur



- https://medium.com/@dslmeinte/what-goes-into-creating-a-projectional-editor-838e8b704018

Projectional Editor: Architektur



- https://medium.com/@dslmeinte/what-goes-into-creating-a-projectional-editor-838e8b704018

Projectional Editing: Vor- und Nachteile

+ Kaum Potential für Syntaxfehler
+ Beliebige Syntax darstellbar
+ Keine Uneindeutigkeit
+ Kombinieren von Sprachen
+ Leichter lesbar
+ Geeignet für Domänenexperten

– Gewöhnungsbedürftig
  – Explizite Konzeptwahl mit Completion
– Syntax muss erlernt werden
– Nichttriviale Toolintegration
  – Git/VCS
  – Diff
  – Suchen/Ersetzen
– Keine Standardformate
  – Teilen von Quellcode im Internet meist durch Bilder

Mehr Infos

- MPS Tutorialreihe: https://www.jetbrains.com/help/mps/fast-track-to-mps.html
- Projectional Editing von Martin Fowler: https://martinfowler.com/bliki/ProjectionalEditing.html