

Merkmalsmodelle mit Linux-KConfig

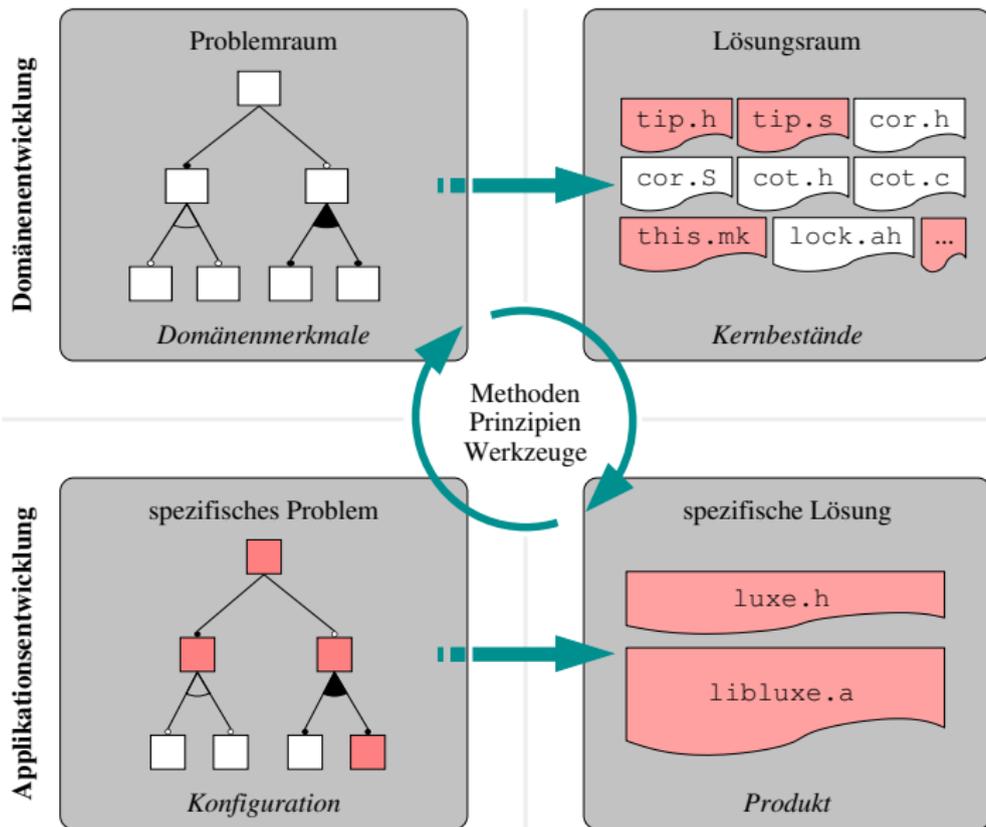
Benjamin Oechslein, Julio Sincero

Friedrich-Alexander-Universität Erlangen-Nürnberg
Lehrstuhl Informatik 4 (Verteilte Systeme und Betriebssysteme)

`www4.informatik.uni-erlangen.de`

6. Juni 2012

Merkmalsmodellierung

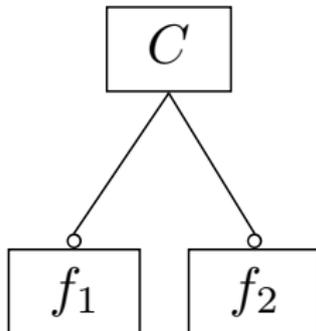


KConfig Syntax

```
config C
    bool "Merkmal 0"
    default y

config f1
    bool "Merkmal 1"
    depends on C
    default n

config f2
    bool
    prompt "Merkmal 2"
    depends on C
    default y
```



```
config C
    bool "Merkmal 0"
    default y

config f1
    bool "Merkmal 1"
    depends on C
    default n

config f2
    bool
    prompt "Merkmal 2"
    depends on C
    default y
```

Mögliche Datentypen

bool mit Werten y,n

tristate wie bool, jedoch
zusätzlich m (Modul)

int mit range x y
einschränkbar

hex wie int, aber in
hexadezimaler
Darstellung

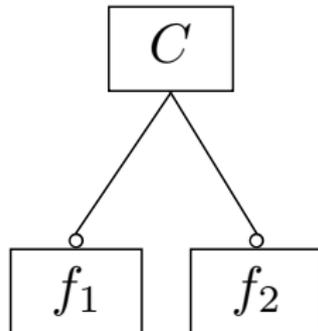
string

KConfig Syntax

```
menuconfig C
    bool "Merkmal 0"
    default y
```

```
config f1
    bool "Merkmal 1"
    depends on C
    default n
```

```
config f2
    bool "Merkmal 2"
    depends on C
    default y
```



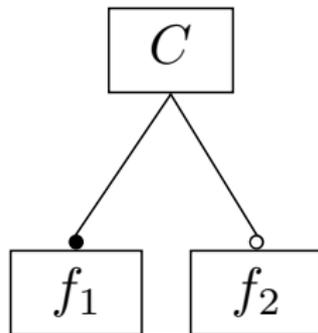
menuconfig sorgt für zusätzliche visuelle Untergliederung

KConfig Syntax

```
config C
    bool "Merkmal 0"
    default y
```

```
config f1
    bool
    depends on C
    default y
comment "Merkmal 1"
    depends on f1
```

```
config f2
    bool "Merkmal 2"
    depends on C
    default y
```

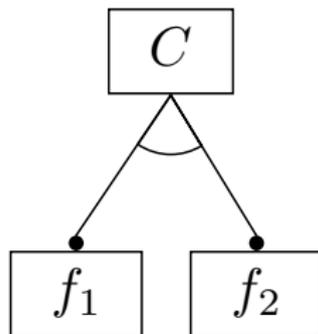


KConfig Syntax

```
choice
prompt "Auswahl"

config A
    bool "A"
config B
    bool "B"

endchoice
```



```
config A
    bool "A"
config B
    bool "B"
    depends on !A
config C
    bool "C"
    select B
```

ACHTUNG: `select` ignoriert mit `depends` getroffene Entscheidungen. Darstellung in `mconf` und tatsächliche Auswahl weichen u.U. ab.

- ▶ Eine angepasste Variante von KConfig befindet sich unter `/proj/i4bst/tools/bst-config/bin`

- ▶ Name der `.config` Datei wird durch Environment-Variable `KCONFIG_CONFIG` gesteuert → Möglichkeit frühere Konfigurationen aufzubewahren!

- ▶ Abbildung von konkreten Konfigurationen auf Dateimengen
 - ▶ Zuordnung von Implementierungsartefakten zu Konfigurationsmerkmalen durch ein Familienmodell
 - ▶ Erzeugung der konfigurierten Quellen durch ein Transformationswerkzeug

- ▶ BST-config bietet hierfür ein Script: `transform.pl`
Aufruf:

```
$> transform.pl -i inputdir -o outputdir \  
                -f config -m familymodel
```

BST-config Familienmodellbeschreibung

```
$cmp={
  dir='',
  name => 'model_example',
  comp => [
    {
      subdir='compA',
      name => 'Component_A',
      depends => '&featA',
      files => [
        "CompA.h",
        "CompA.cc",
      ],
    },
    {
      subdir='compB',
      name => 'Component_B',
      depends => '&featB',
      files => [
        "CompB.h",
        "CompB.cc",
      ],
    },
  ],
};
```