

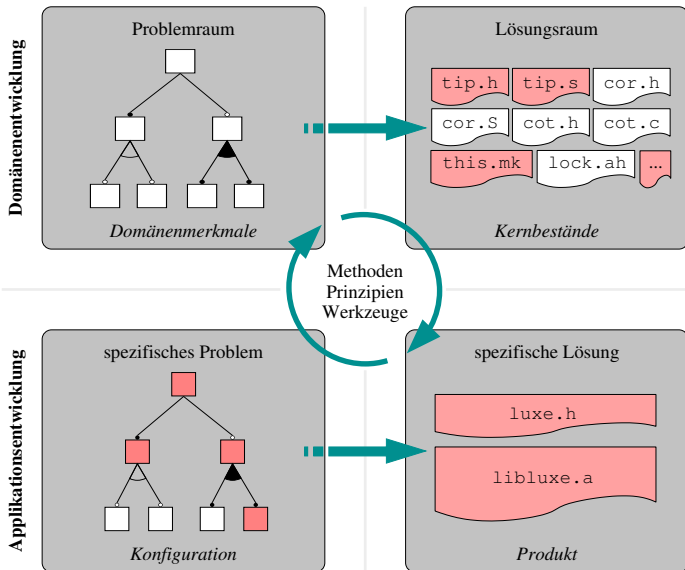
Merkmalsmodelle mit Linux-KConfig

Daniel Danner, Gabor Drescher

Friedrich-Alexander-Universität Erlangen-Nürnberg
Lehrstuhl Informatik 4 (Verteilte Systeme und Betriebssysteme)
`www4.informatik.uni-erlangen.de`

May 22, 2014





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

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

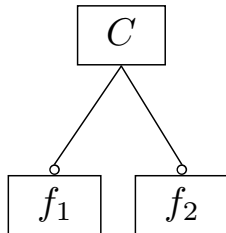
```
config f2
    bool
    prompt "Merkmal 2"
    depends on C
    default y
```



```
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
```



```
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**



```
menuconfig 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
```

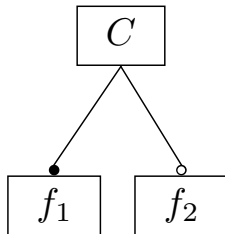
menuconfig sorgt für zusätzliche visuelle Untergliederung



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

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

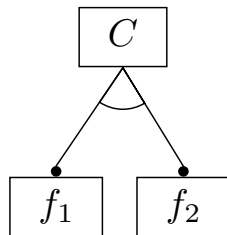
```
config f2
    bool "Merkmal 2"
    depends on C
    default y
```



```
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 im Vorgabe-Tarball
- 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
- Die Vorgabe enthält hierfür ein Script: `transform.pl`
Aufruf:

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



```
$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",  
      ],  
    },  
  ],  
};
```

