# Worum geht es?

- GUI mit generischer Behandlung von Properties
- Anpassbare Darstellung (Datenreihen → Bilder, etc.)
- Zusatzfunktionen zu Geräten

- Nutzung der Metadaten aus den XMLs

# Worum geht es?

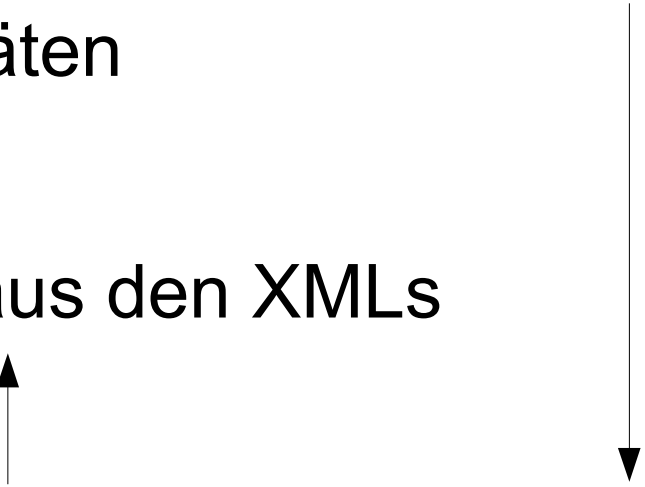
- GUI mit generischer Behandlung von Properties
- Anpassbare Darstellung (Datenreihen → Bilder, etc.)
- Zusatzfunktionen zu Geräten

- Nutzung der Metadaten aus den XMLs

- XML-Struktur Vorraussetzung für “nicht-flache” Darst.
- => Hierarchischen Zugriff auf Properties auslagern
- => Auch Verwendung aus interaktivem Interpreter

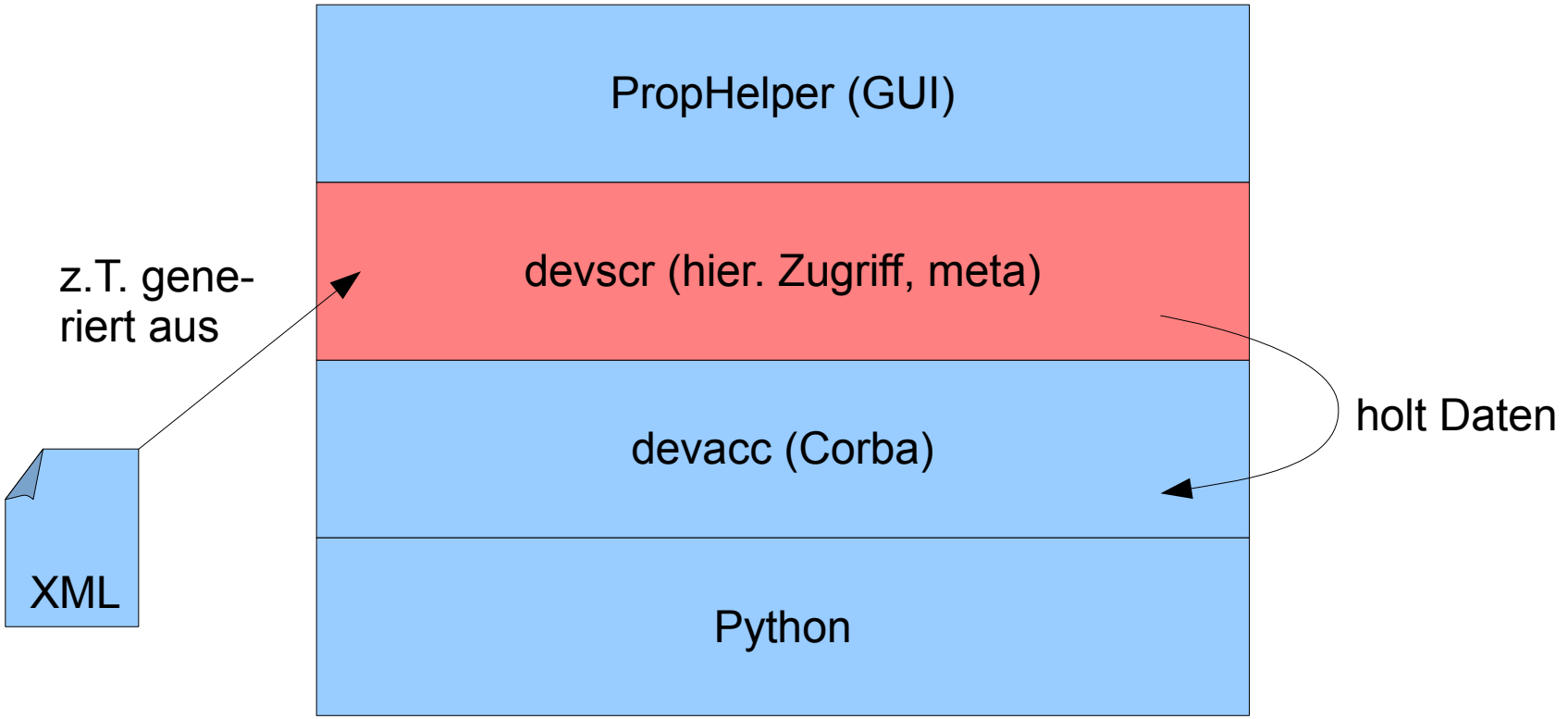
# Worum geht es?

- GUI mit generischer Behandlung von Properties
- Anpassbare Darstellung (Datenreihen → Bilder, etc.)
- Zusatzfunktionen zu Geräten



- Nutzung der Metadaten aus den XMLs
- XML-Struktur Vorraussetzung für “nicht-flache” Darst.
- => Hierarchischen Zugriff auf Properties auslagern
- => Auch Verwendung aus interaktivem Interpreter

# Worum geht es?



# devscr: erster Kontakt

- Generierung von Quellcode aus XMLs
  => Namen, Typen, Beschreibungen, Hierarchie

- Automatische Geräte-Erzeugung (z.B. Typ MX)
- Inspektion auf Gerät- und Datenebene
- unabhängige Property-Objekte

# devscr: erster Kontakt

- Generierung von Quellcode aus XMLs
  => Namen, Typen, Beschreibungen, Hierarchie

- Automatische Objekterzeugung (z.B. Typ MX)
- Inspektion auf Gerät- und Datenebene
- unabhängige Property-Objekte

```
% python
>>> import devscr
>>>
```

# devscr: erster Kontakt

- Generierung von Quellcode aus XMLs
  => Namen, Typen, Beschreibungen, Hierarchie

- Automatische Objekterzeugung (z.B. Typ MX)
- Inspektion auf Gerät- und Datenebene
- unabhängige Property-Objekte

```
% python
>>> import devscr
>>> u=devscr.create(“ut1mk1”)
>>>
```

# devscr: erster Kontakt

- Generierung von Quellcode aus XMLs
  => Namen, Typen, Beschreibungen, Hierarchie

- Automatische Objekterzeugung (z.B. Typ MX)
- Inspektion auf Gerät- und Datenebene
- unabhängige Property-Objekte

```
% python
>>> import devscr
>>> u=devscr.create(“ut1mk1”)
>>> print u
UT1MK1 (MXEqMod)
>>>
```

# devscr: erster Kontakt

- Generierung von Quellcode aus XMLs
  => Namen, Typen, Beschreibungen, Hierarchie

- Automatische Objekterzeugung (z.B. Typ MX)
- Inspektion auf Gerät- und Datenebene
- unabhängige Property-Objekte

```
% python
>>> import devscr
>>> u=devscr.create(“ut1mk1”)
>>> print u
UT1MK1 (MXEqMod)
>>> print u.props
['cinit', 'creset', 'rconstant', 'rmagninfo', ... 'wshutup']
>>>
```

# devscr: erster Kontakt

- Generierung von Quellcode aus XMLs
  => Namen, Typen, Beschreibungen, Hierarchie

- Automatische Objekterzeugung (z.B. Typ MX)
- Inspektion auf Gerät- und Datenebene
- unabhängige Property-Objekte

```
% python
>>> import devscr
>>> u=devscr.create(“ut1mk1”)
>>> print u
UT1MK1 (MXEqMod)
>>> print u.props
['cinit', 'creset', 'rconstant', 'rmagninfo', ... 'wshutup']
>>> c=u.rconstant()
>>>
```

# devscr: erster Kontakt

- Generierung von Quellcode aus XMLs
  => Namen, Typen, Beschreibungen, Hierarchie

- Automatische Objekterzeugung (z.B. Typ MX)
- Inspektion auf Gerät- und Datenebene
- unabhängige Property-Objekte

```
% python
>>> import devscr
>>> u=devscr.create(“ut1mk1”)
>>> print u
UT1MK1 (MXEqMod)
>>> print u.props
['cinit', 'creset', 'rconstant', 'rmagninfo', ... 'wshutup']
>>> c=u.rconstant()
>>> print c.data.items
['mincurrent', 'maxcurrent', 'minhysteresis', 'subtype', ...]
>>>
```

# devscr: Hierarchie

```
<property category="master">
<name>CONSTANT</name>
<description>
        <short> Read device constants. </short>
        <long>[...]
        </long>
</description>
<action type="read" access="free" medlock="none"/>
<data>
        <value type="Float32" name="minCurrent">
        Minimum current set value in A.</value>
        <value type="Float32" name="maxCurrent">
        Maximum current set value in A.</value>
        <value type="Float32" name="minHysteresis">
        Minimum current set value in A for hysteresis
        corrections.</value>
        <value type="Float32" name="subtype">[...]</value>

        <array length="3" ref="Polynomial3" name="polyBII">
        Set of 3 polynomials of 3rd order to calculate \f$BI(I)\f$.
        </array>
        <array length="3" ref="Polynomial3" name="polyIBI">
        Set of 3 polynomials of 3rd order to calculate \f$I(BI)\f$.
        </array>
        [ … noch zwei arrays … ]
</data>
</property>
```

- Namen aus XMLs
- Arrays mit Indices
- Zusammenge-
  setzte Typen...
- … mit/in Arrays

```
<complextype name="Polynomial3" type="Float32">
<description>
        <short>Structure of one polynomial of the device
        constants.</short>
</description>
<value name="start">Start of valid polynomial range.</value>
<value name="end">End of valid polynomial range.</value>
<value name="a0">Coefficient \f$a_0\f$.</value>
<value name="a1">Coefficient \f$a_1\f$.</value>
<value name="a2">Coefficient \f$a_2\f$.</value>
<value name="a3">Coefficient \f$a_3\f$.</value>
</complextype>
```

u UT1MK1 (MXEqMod)

c

# devscr: Hierarchie

```
<property category="master">
<name>CONSTANT</name>
<description>
        <short> Read device constants. </short>
        <long>[...]
        </long>
</description>
<action type="read" access="free" medlock="none"/>
<data>
        <value type="Float32" name="minCurrent">
        Minimum current set value in A.</value>
        <value type="Float32" name="maxCurrent">
        Maximum current set value in A.</value>
        <value type="Float32" name="minHysteresis">
        Minimum current set value in A for hysteresis
        corrections.</value>
        <value type="Float32" name="subtype">[...]</value>

        <array length="3" ref="Polynomial3" name="polyBII">
        Set of 3 polynomials of 3rd order to calculate \f$BI(I)\f$.
        </array>
        <array length="3" ref="Polynomial3" name="polyIBI">
        Set of 3 polynomials of 3rd order to calculate \f$I(BI)\f$.
        </array>
        [ … noch zwei arrays … ]
</data>
</property>
```

```
<complextype name="Polynomial3" type="Float32">
<description>
        <short>Structure of one polynomial of the device
        constants.</short>
</description>
<value name="start">Start of valid polynomial range.</value>
<value name="end">End of valid polynomial range.</value>
<value name="a0">Coefficient \f$a_0\f$.</value>
<value name="a1">Coefficient \f$a_1\f$.</value>
<value name="a2">Coefficient \f$a_2\f$.</value>
<value name="a3">Coefficient \f$a_3\f$.</value>
</complextype>
```

```
>>> c=u.rconstant()
>>> print c.data.maxcurrent
1000.0
```

u UT1MK1 (MXEqMod)

c

# devscr: Hierarchie

```
<property category="master">
<name>CONSTANT</name>
<description>
        <short> Read device constants. </short>
        <long>[...]
        </long>
</description>
<action type="read" access="free" medlock="none"/>
<data>
        <value type="Float32" name="minCurrent">
        Minimum current set value in A.</value>
        <value type="Float32" name="maxCurrent">
        Maximum current set value in A.</value>
        <value type="Float32" name="minHysteresis">
        Minimum current set value in A for hysteresis
        corrections.</value>
        <value type="Float32" name="subtype">[...]</value>

        <array length="3" ref="Polynomial3" name="polyBII">
        Set of 3 polynomials of 3rd order to calculate \f$BI(I)\f$.
        </array>
        <array length="3" ref="Polynomial3" name="polyIBI">
        Set of 3 polynomials of 3rd order to calculate \f$I(BI)\f$.
        </array>
        [ … noch zwei arrays … ]
</data>
</property>
```

```
<complextype name="Polynomial3" type="Float32">
<description>
        <short>Structure of one polynomial of the device
        constants.</short>
</description>
<value name="start">Start of valid polynomial range.</value>
<value name="end">End of valid polynomial range.</value>
<value name="a0">Coefficient \f$a_0\f$.</value>
<value name="a1">Coefficient \f$a_1\f$.</value>
<value name="a2">Coefficient \f$a_2\f$.</value>
<value name="a3">Coefficient \f$a_3\f$.</value>
</complextype>
```

```
>>> c=u.rconstant()
>>> print c.data.maxcurrent
1000.0
>>> print len(c.data.polyibl)
3
```

u UT1MK1 (MXEqMod)

c

# devscr: Hierarchie

```
<property category="master">
<name>CONSTANT</name>
<description>
        <short> Read device constants. </short>
        <long>[...]
        </long>
</description>
<action type="read" access="free" medlock="none"/>
<data>
        <value type="Float32" name="minCurrent">
        Minimum current set value in A.</value>
        <value type="Float32" name="maxCurrent">
        Maximum current set value in A.</value>
        <value type="Float32" name="minHysteresis">
        Minimum current set value in A for hysteresis
        corrections.</value>
        <value type="Float32" name="subtype">[...]</value>

        <array length="3" ref="Polynomial3" name="polyBII">
        Set of 3 polynomials of 3rd order to calculate \f$BI(I)\f$.
        </array>
        <array length="3" ref="Polynomial3" name="polyIBI">
        Set of 3 polynomials of 3rd order to calculate \f$I(BI)\f$.
        </array>
        [ … noch zwei arrays … ]
</data>
</property>
```

2

```
<complextype name="Polynomial3" type="Float32">
<description>
        <short>Structure of one polynomial of the device
        constants.</short>
</description>
<value name="start">Start of valid polynomial range.</value>
<value name="end">End of valid polynomial range.</value>
<value name="a0">Coefficient \f$a_0\f$.</value>
<value name="a1">Coefficient \f$a_1\f$.</value>
<value name="a2">Coefficient \f$a_2\f$.</value>
<value name="a3">Coefficient \f$a_3\f$.</value>
</complextype>
```

```
>>> c=u.rconstant()
>>> print c.data.maxcurrent
1000.0
>>> print len(c.data.polyibl)
3
>>> print c.data.polyibl[2]
start : 0.303429991007
end   : 0.433450013399
a0    : 3.22249388695
a1    : -2334.04003906
a2    : 89.4224700928
a3    : -90.0511322021
```

u UT1MK1 (MXEqMod)

c

# devscr: Hierarchie

```
<property category="master">
<name>CONSTANT</name>
<description>
        <short> Read device constants. </short>
        <long>[...]
        </long>
</description>
<action type="read" access="free" medlock="none"/>
<data>
        <value type="Float32" name="minCurrent">
        Minimum current set value in A.</value>
        <value type="Float32" name="maxCurrent">
        Maximum current set value in A.</value>
        <value type="Float32" name="minHysteresis">
        Minimum current set value in A for hysteresis
        corrections.</value>
        <value type="Float32" name="subtype">[...]</value>

        <array length="3" ref="Polynomial3" name="polyBII">
        Set of 3 polynomials of 3rd order to calculate \f$BI(I)\f$.
        </array>
        <array length="3" ref="Polynomial3" name="polyIBI">
        Set of 3 polynomials of 3rd order to calculate \f$I(BI)\f$.
        </array>
        [ … noch zwei arrays … ]
</data>
</property>
```

2

```
<complextype name="Polynomial3" type="Float32">
<description>
        <short>Structure of one polynomial of the device
        constants.</short>
</description>
<value name="start">Start of valid polynomial range.</value>
<value name="end">End of valid polynomial range.</value>
<value name="a0">Coefficient \f$a_0\f$.</value>
<value name="a1">Coefficient \f$a_1\f$.</value>
<value name="a2">Coefficient \f$a_2\f$.</value>
<value name="a3">Coefficient \f$a_3\f$.</value>
</complextype>
```

```
>>> c=u.rconstant()
>>> print c.data.maxcurrent
1000.0
>>> print len(c.data.polyibl)
3
>>> print c.data.polyibl[2]
start : 0.303429991007
end   : 0.433450013399
a0    : 3.22249388695
a1    : -2334.04003906
a2    : 89.4224700928
a3    : -90.0511322021
>>> print c.data.polyibl[2].start
0.303429991007

>>> print u.read("constant")[34]
0.303429991007
```

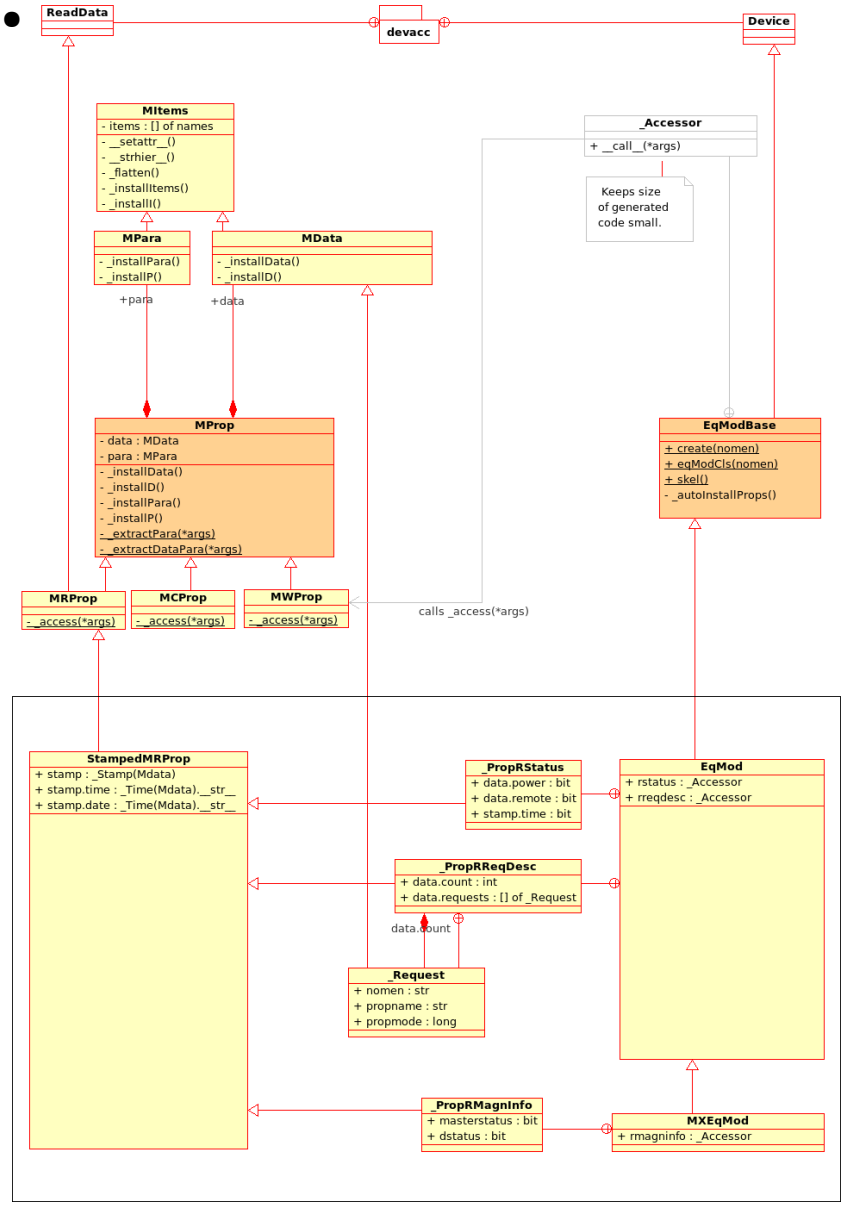
# devscr: generierter Code

```
<property category="master">
<name>CONSTANT</name>
<description>
        <short> Read device constants. </short>
        <long>[...]
        </long>
</description>
<action type="read" access="free" medlock="none"/>
<data>
        <value type="Float32" name="minCurrent">
        Minimum current set value in A.</value>
        <value type="Float32" name="maxCurrent">
        Maximum current set value in A.</value>
        <value type="Float32" name="minHysteresis">
        Minimum current set value in A for hysteresis
        corrections.</value>
        <value type="Float32" name="subtype">[...]</value>

        <array length="3" ref="Polynomial3" name="polyBII">
        Set of 3 polynomials of 3rd order to calculate \f$BI(I)\f$.
        </array>
        <array length="3" ref="Polynomial3" name="polyIBI">
        Set of 3 polynomials of 3rd order to calculate \f$I(BI)\f$.
        </array>
        [ … noch zwei arrays … ]
</data>
</property>
```

```
<complextype name="Polynomial3" type="Float32">
<description>
        <short>Structure of one polynomial of the device
        constants.</short>
</description>
<value name="start">Start of valid polynomial range.</value>
<value name="end">End of valid polynomial range.</value>
<value name="a0">Coefficient \f$a_0\f$.</value>
<value name="a1">Coefficient \f$a_1\f$.</value>
<value name="a2">Coefficient \f$a_2\f$.</value>
<value name="a3">Coefficient \f$a_3\f$.</value>
</complextype>
```

- gen. Code (mögl.) unabh. von Impl.
-



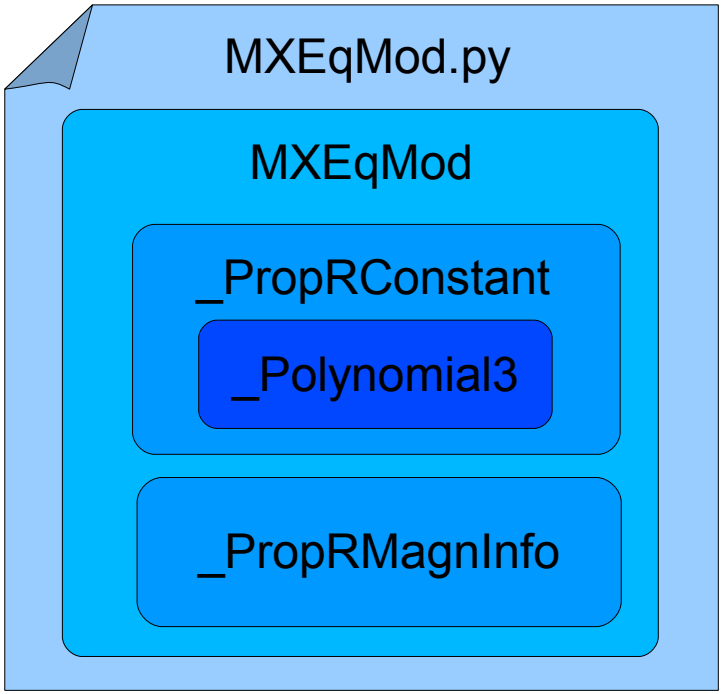
# devscr: generierter Code

```
<property category="master">
<name>CONSTANT</name>
<description>
        <short> Read device constants. </short>
        <long>[...]
        </long>
</description>
<action type="read" access="free" medlock="none"/>
<data>
        <value type="Float32" name="minCurrent">
        Minimum current set value in A.</value>
        <value type="Float32" name="maxCurrent">
        Maximum current set value in A.</value>
        <value type="Float32" name="minHysteresis">
        Minimum current set value in A for hysteresis
        corrections.</value>
        <value type="Float32" name="subtype">[...]</value>

        <array length="3" ref="Polynomial3" name="polyBII">
        Set of 3 polynomials of 3rd order to calculate \f$BI(I)\f$.
        </array>
        <array length="3" ref="Polynomial3" name="polyIBI">
        Set of 3 polynomials of 3rd order to calculate \f$I(BI)\f$.
        </array>
        [ … noch zwei arrays … ]
</data>
</property>
```

```
<complextype name="Polynomial3" type="Float32">
<description>
        <short>Structure of one polynomial of the device
        constants.</short>
</description>
<value name="start">Start of valid polynomial range.</value>
<value name="end">End of valid polynomial range.</value>
<value name="a0">Coefficient \f$a_0\f$.</value>
<value name="a1">Coefficient \f$a_1\f$.</value>
<value name="a2">Coefficient \f$a_2\f$.</value>
<value name="a3">Coefficient \f$a_3\f$.</value>
</complextype>
```

- gen. Code (mögl.) unabh. von Impl.
- Gerätemodul (eine Datei)
  → Geräteklasse (genau eine)
  → Propertyklassen
  → Datentypklassen
  → ...
-




- _installData() wird von devscr gerufen
- _installD() macht die Arbeit

# devscr: generierter Code

```
<property category="master">
<name>CONSTANT</name>
<description>
        <short> Read device constants. </short>
        <long>[...]
        </long>
</description>
<action type="read" access="free" medlock="none"/>
<data>
        <value type="Float32" name="minCurrent">
        Minimum current set value in A.</value>
        <value type="Float32" name="maxCurrent">
        Maximum current set value in A.</value>
        <value type="Float32" name="minHysteresis">
        Minimum current set value in A for hysteresis
        corrections.</value>
        <value type="Float32" name="subtype">[...]</value>

        <array length="3" ref="Polynomial3" name="polyBIl">
        Set of 3 polynomials of 3rd order to calculate \f$Bl(I)\f$.
        </array>
        <array length="3" ref="Polynomial3" name="polyIBl">
        Set of 3 polynomials of 3rd order to calculate \f$I(Bl)\f$.
        </array>
        [ … noch zwei arrays … ]
</data>
</property>
```

```
<complextype name="Polynomial3" type="Float32">
<description>
        <short>Structure of one polynomial of the device
        constants.</short>
</description>
<value name="start">Start of valid polynomial range.</value>
<value name="end">End of valid polynomial range.</value>
<value name="a0">Coefficient \f$a_0\f$.</value>
<value name="a1">Coefficient \f$a_1\f$.</value>
<value name="a2">Coefficient \f$a_2\f$.</value>
<value name="a3">Coefficient \f$a_3\f$.</value>
</complextype>
```

```
% cat MXEqMod.py

[ … imports … ]



class MXEqMod:

    [ … andere Properties … ]



    class _PropRConstant(StampedMRProp):

        PROPNAME = "constant"
        META = "Read device constants."

        class _Polynomial3(MData):

            META = "Structure of one polynomial of the device constants."

            def _installData(self, i):
                self._installD("start", float(i.next()), "Start of valid
poly...
                self._installD("end", float(i.next()), "End of valid poly...
                self._installD("a0", float(i.next()), "Coefficient
\f$a_0\f$.")
                self._installD("a1", float(i.next()), "Coefficient
\f$a_1\f$.")
                self._installD("a2", float(i.next()), "Coefficient
\f$a_2\f$.")
                self._installD("a3", float(i.next()), "Coefficient
\f$a_3\f$.")

        def _installData(self, i = Mitems.ZEROIT):
            self._installD("mincurrent", float(i.next()), "Minimum cur...
            self._installD("maxcurrent", float(i.next()), "Maximum...
            self._installD("minhysteresis", float(i.next()), "Minimum cu...
            self._installD("subtype", float(i.next()), "Device subtype […]")

            arr = [self._Polynomial3(i) for j in range(3)]
            self._installD("polybli", MData(iter(arr)), "Set of 3 polynomi...
            arr = [self._Polynomial3(i) for j in range(3)]
            self._installD("polyibl", MData(iter(arr)), "Set of 3 po...



        [ … andere Properties … ]
```

# devscr: generierter Code

- Ändere _installData() für Abb. nach speziellen Bedürfnissen
- z.B. Bitmasken in Bits zerlegen und den Bits Namen zuweisen
- >>> print u.rstatus().data.magnet1
  1

```
% cat MXEqMod.py

[ … imports … ]


class MXEqMod:

    [ … andere Properties … ]


    class _PropR Constant(StampedMRProp):

        PROPNAME = "constant"
        META = "Read device constants."

        class _Polynomial3(MData):

            META = "Structure of one polynomial of the device constants."

            def _installData(self, i):
                self._installD("start", float(i.next()), "Start of valid
poly...

                self._installD("end", float(i.next()), "End of valid poly...
                self._installD("a0", float(i.next()), "Coefficient
\f$a_0\f$.")
                self._installD("a1", float(i.next()), "Coefficient
\f$a_1\f$.")
                self._installD("a2", float(i.next()), "Coefficient
\f$a_2\f$.")
                self._installD("a3", float(i.next()), "Coefficient
\f$a_3\f$.")

        def _installData(self, i = Mitems.ZEROIT):
            self._installD("mincurrent", float(i.next()), "Minimum cur...
            self._installD("maxcurrent", float(i.next()), "Maximum...
            self._installD("minhysteresis", float(i.next()), "Minimum cu...
            self._installD("subtype", float(i.next()), "Device subtype […]")

            arr = [self._Polynomial3(i) for j in range(3)]
            self._installD("polybli", MData(iter(arr)), "Set of 3 polynomi...
            arr = [self._Polynomial3(i) for j in range(3)]
            self._installD("polyibl", MData(iter(arr)), "Set of 3 po...


    [ … andere Properties … ]
```

# Descriptions/meta?

- Werden nur bei Bedarf/auf Befehl geladen.
- Nur für benötigte Geräte.
- Einmal pro Property-/Datenklasse im Speicher.

# Descriptions/meta?

- Werden nur bei Bedarf/auf Befehl geladen.
- Nur für benötigte Geräte.
- Einmal pro Property-/Datenklasse im Speicher.

- ```
  % python
  >>> import devscr
  >>> u=devscr.create(“ut1mk1”)
  >>> print u.rconstant().data.meta(“mincurrent”)
  ?
  ```
  ```

# Descriptions/meta?

- Werden nur bei Bedarf/auf Befehl geladen.
- Nur für benötigte Geräte.
- Einmal pro Property-/Datenklasse im Speicher.

- ```
  % python
  >>> import devscr
  >>> u=devscr.create(“ut1mk1”)
  >>> print u.rconstant().data.meta(“mincurrent”)
  ?
  >>> devscr.installMeta(“ut1mk1”) # oder (“mx”)
  >>> print u.rconstant().data.meta(“mincurrent”)
  Minimum current set value in A.
  ```
  ```