



Ruhr-Universität Bochum
Fakultät für Bau- und Umweltingenieurwissenschaften
Lehrstuhl für Hydrologie, Wasserwirtschaft und Umwelttechnik
Prof. Dr. rer. nat. habil. A. Schumann



Weiterentwicklung hydrologischer Modellsysteme im Hinblick auf Transparenz, Flexibilität und Qualitätssicherung am Beispiel des Modell-Frameworks HydPy und dem WHM LARSIM

Christoph Tyralla (RUB → BCE)
Gordon Horn (RUB)
Gernot Belger (BCE)
Peter Krahe (BfG)



Bundesanstalt für Gewässerkunde
Referat M2: Wasserhaushalt, Vorhersagen und Prognosen

“Most computational hydrology is not reproducible, so is it really science?”

Christopher Hutton, Thorsten Wagener, Jim Freer, Dawei Han, Chris Duffy, Berit Arheimer, WRR 2016

“Bit-Reproducibility vs. Conclusion Reproducibility”

Lieke A. Melsen, Paul J.J.F. Torfs, Remko Uijlenhoet, Adriaan J. Teuling, WRR 2017

“Insufficient computational details and ignorance of copyright laws”

Juan A. Anel, WRR 2017

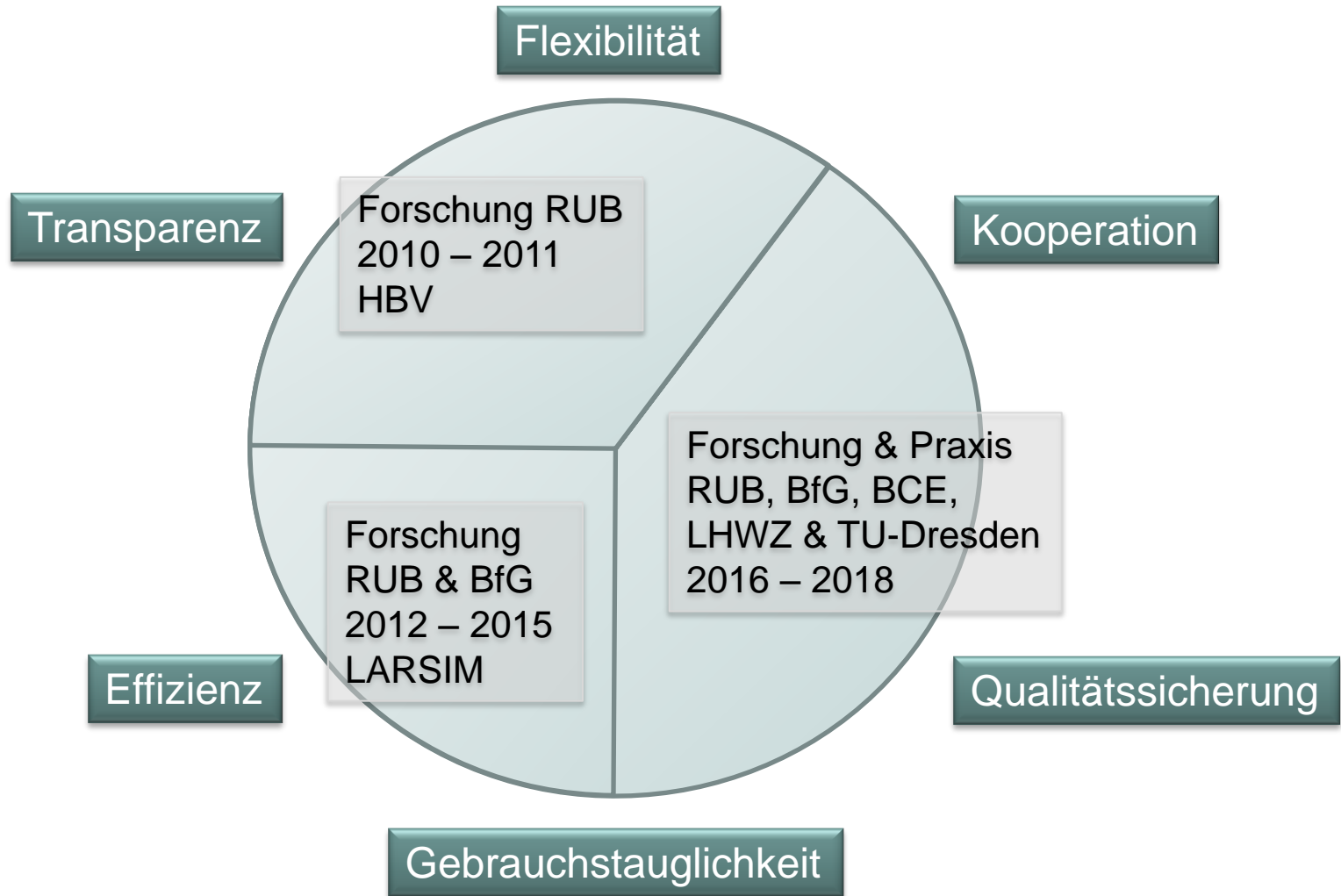
“Cultural change that is the greatest challenge”

Christopher Hutton, Thorsten Wagener, Jim Freer, Dawei Han, Chris Duffy, Berit Arheimer, WRR 2017

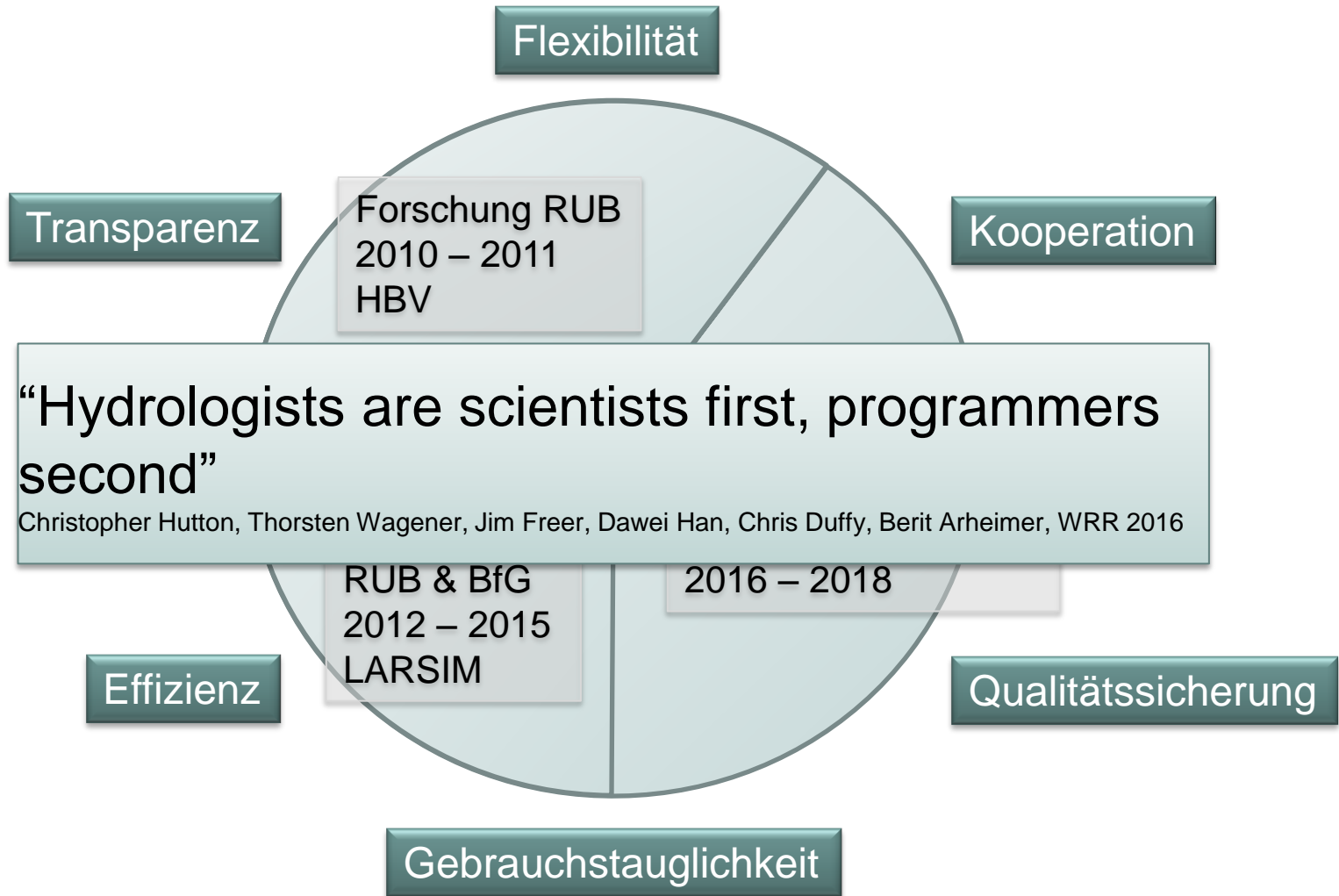
“Use latest best practices in designing research software”

R.W. Hut, N.C. van de Giesen, N. Drost, WRR 2017

Entwicklungshistorie HydPy



Entwicklungshistorie HydPy



Programmiersprache

- CPython:
 - Open Source
 - leichter Einstieg
 - viele (wissenschaftliche) Bibliotheken
 - Eignung für kurze Skripte und komplexe Programme
 - dynamische Typisierung (langsam)
- Cython:
 - Open Source
 - Metasprache für Python und C
 - Interaktion von Python mit C, C++ und Fortran
 - statische Typisierung (schnell)
- Umsetzung in HydPy:
 - Frameworkprogrammierung (primär) in Python
 - Modellprogrammierung in Python
 - Automatische Übersetzung nach Cython
 - Automatische Übersetzung nach C
 - Automatisches Kompilieren und Einbinden in Python
 - Modellkonfiguration in Python



Modellkonfiguration über Textdateien (Beispiel LARSIM Tape12)

```

* 2436 Profilart: Gerinneschaetzer (Stand: Juli 2005)
* gesamtes Einzugsgebiet [qkm]      =    1148.72
* Gerinnebildender Abfluss [cbm/s] =    221.56
2436 ID_008611    26.723    2.58    596.00    448.00    3942.500    2317.500    1
2436 1074207    1064601    0.00609    2.23    24.07    0.00    0.00    2
2436 24.07    24.07    2.11    100.00    100.00    4.00    4.00    3a
2436 29.23    20.00    20.00
2436 3.8 0.0 1.2 43.5 0.0 0.0 0.0 0.0 27.0 0.0 0.0 1.5 0.0 0.0 4a
2436 23.0 0.0
2436 150. 0. 100. 176. 0. 0. 0. 0. 172. 0. 0. 153. 0. 0. 5a
2436 163. 0.
*

```

===== 1. Zeile pro Element (Datenkarte 10) =====

I5	A10	F10.3	F10.3	F10.3	F10.3	F10.3	F10.3	I5
Ele- ment	Flächen- nummer	Flächen- größe	Vorfluter Länge	Vorfluter Höhe ob.	Vorfluter Höhe un.	Rechts- wert	Hoch- wert	Zeile
J	NRFLV	FT	TAL	HOT	HUT	X	Y	J1
[-]	[-]	[qkm]	[km]	[m+NN]	[m+NN]	[km]	[km]	[1]

Modellkonfiguration über Quelltextdateien (Beispiel control-Datei)

```
# -*- coding: utf-8 -*-  
  
from hydp.models.lland_v1 import *  
  
parameterstep("1d")  
simulationstep("1h")  
  
ft(26.808)  
nhru(7)  
lnk(VERS, ACKER, GRUE_I, BAUMB, NADELW, LAUBW, MISCHW)  
nfk(acker=92.0, baumb=82.0, grue_i=88.0, laubw=84.0, mischw=88.0,  
     nadelw=88.0, vers=0.0)  
fln(auxfile="landuse_month_parameters")  
eqb(2000.0)  
eqi1(1000.0)  
eqi2(500.0)  
eqd1(250.0)  
eqd2(125.0)  
...
```

Modellkonfiguration über Quelltextdateien (Beispiel control-Datei)

```
# -*- coding: utf-8 -*-  
  
from hydp.models.lland_v1 import *  
  
parameterstep("1d")  
simulationstep("1h")  
  
ft(26.808)  
nhru(7)  
lnk(VERS, ACKER, GRUE_I, BAUMB, NADELW, LAUBW, MISCHW)  
nfk(acker=92.0, baumb=82.0, grue_i=88.0, laubw=84.0, mischw=88.0,  
     nadelw=88.0, vers=0.0)  
fln(auxfile="landuse_month_parameters")  
eqb(2000.0)  
eqi1(eqb/2)  
eqi2(eqi1/2)  
eqd1(eqi2/2)  
eqd2(eqd1/2)  
...
```


Modellkonfiguration über Quelltextdateien (Beispiel control-Datei)

```
# -*- coding: utf-8 -*-

from hydropy.models.lland_v1 import *

parameterstep("1d")
simulationstep("1h")

ft(26.808)
nhru(7)
lnk(VERS, ACKER, GRUE_I, BAUMB, NADELW, LAUBW, MISCHW)
nfk(acker=92.0, baumb=82.0, grue_i=88.0, laubw=84.0, mischw=88.0,
    nadelw=88.0, vers=0.0)
fln(auxfile="landuse_month_parameters")
value = 2000.0
for par in (eqb, eqi1, eqi2, eqd1, eqd2):
    par(value)
    value /= 2
...
```

Modellprogrammierung in Python

```
class FT(parametertools.SingleParameter):  
    """Teileinzugsgebietsfläche (subbasin area) [km²]."""  
    NDIM, TYPE, TIME, SPAN = 0, float, None, (1e-10, None)
```

Modellprogrammierung in Python

```
class NHRU(parametertools.SingleParameter):
    """Anzahl der Hydrotope (number of hydrological response units) [-].
    ...
    """
    NDIM, TYPE, TIME, SPAN = 0, int, None, (1, None)

    def __call__(self, *args, **kwargs):
        parametertools.SingleParameter.__call__(self, *args, **kwargs)
        for subpars in self.subpars.pars.model.parameters:
            for par in subpars:
                if par.NDIM == 1:
                    par.shape = self.value
        for subseqs in self.subpars.pars.model.sequences:
            for seq in subseqs:
                if (seq.NDIM == 1) and (seq.name != "moy"):
                    seq.shape = self.value
```

Transparenz und Kooperation

Fork me on GitHub

- Lizenz LGPL3:
 - Quelltext darf weitergereicht werden
 - Auch bei einer Modifikation bleibt diese Regel aufrecht
 - Verlinken anderer Software mit anderer Lizenz ist aber erlaubt
- GitHub:
 - Online-Plattform zur gemeinschaftlichen Pflege von Quelltext
 - Verteiltes Versionskontrollsystem (Git)
 - Enthält Diskussionsforum, Wiki usw.

Transparenz und Kooperation

Fork me on GitHub

756 lines (623 sloc) | 26.5 KB

Raw

Blame

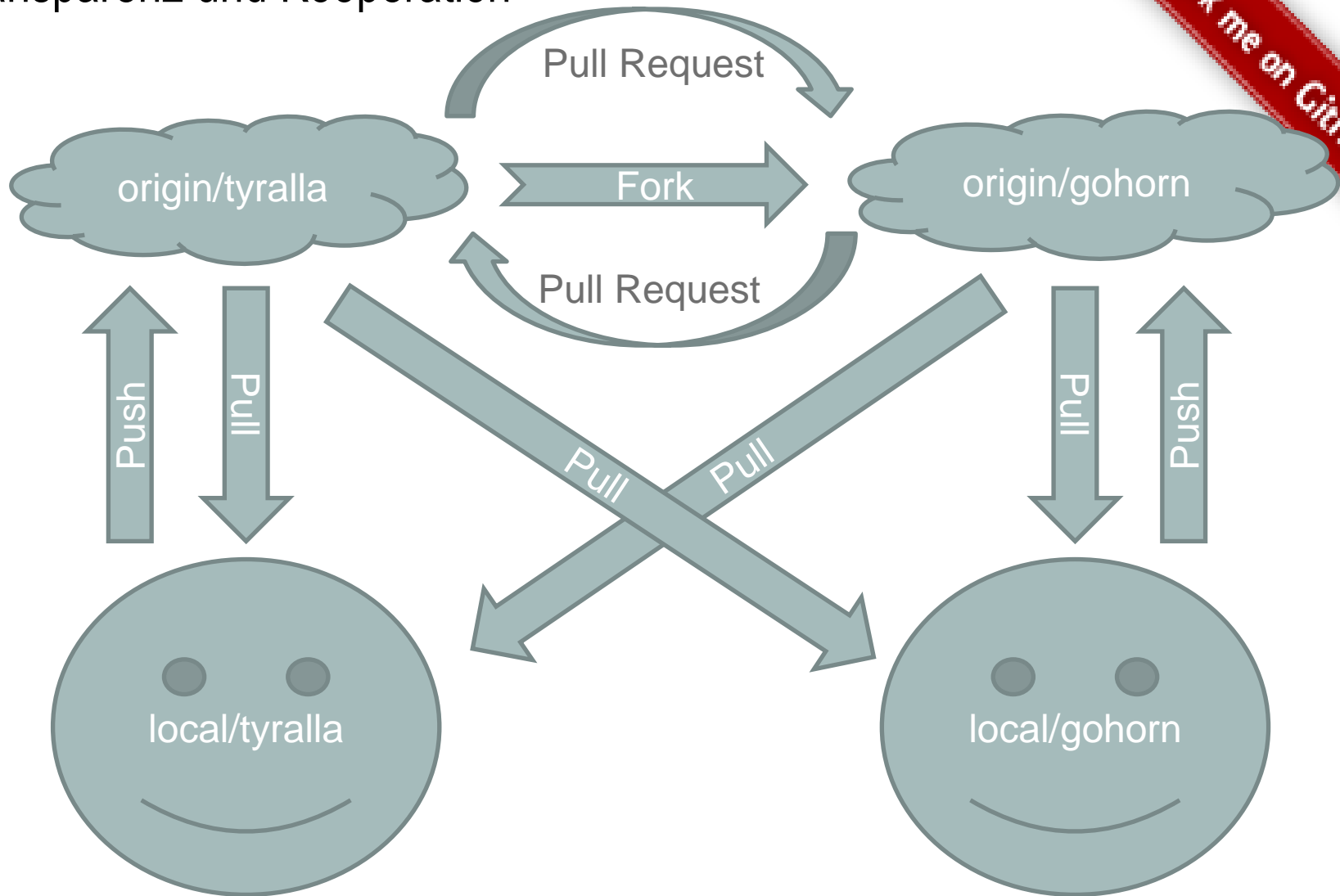
History



```
1  # -*- coding: utf-8 -*-
2
3  # import...
4  # ...from standard library
5  from __future__ import division, print_function
6  import warnings
7  # ...from site-packages
8  import numpy
9  # ...HydPy specific
10 from hydpy.core import parametertools
11 from hydpy.core import objecttools
12 from hydpy.core import timetools
13 # ...model specific
14 from hydpy.models.lland import lland_constants
15 from hydpy.models.lland import lland_parameters
16
17
18 class FT(parametertools.SingleParameter):
19     """Teileinzugsgebietsfläche (subbasin area) [km²]."""
20     NDIM, TYPE, TIME, SPAN = 0, float, None, (1e-10, None)
21
22 ..
```

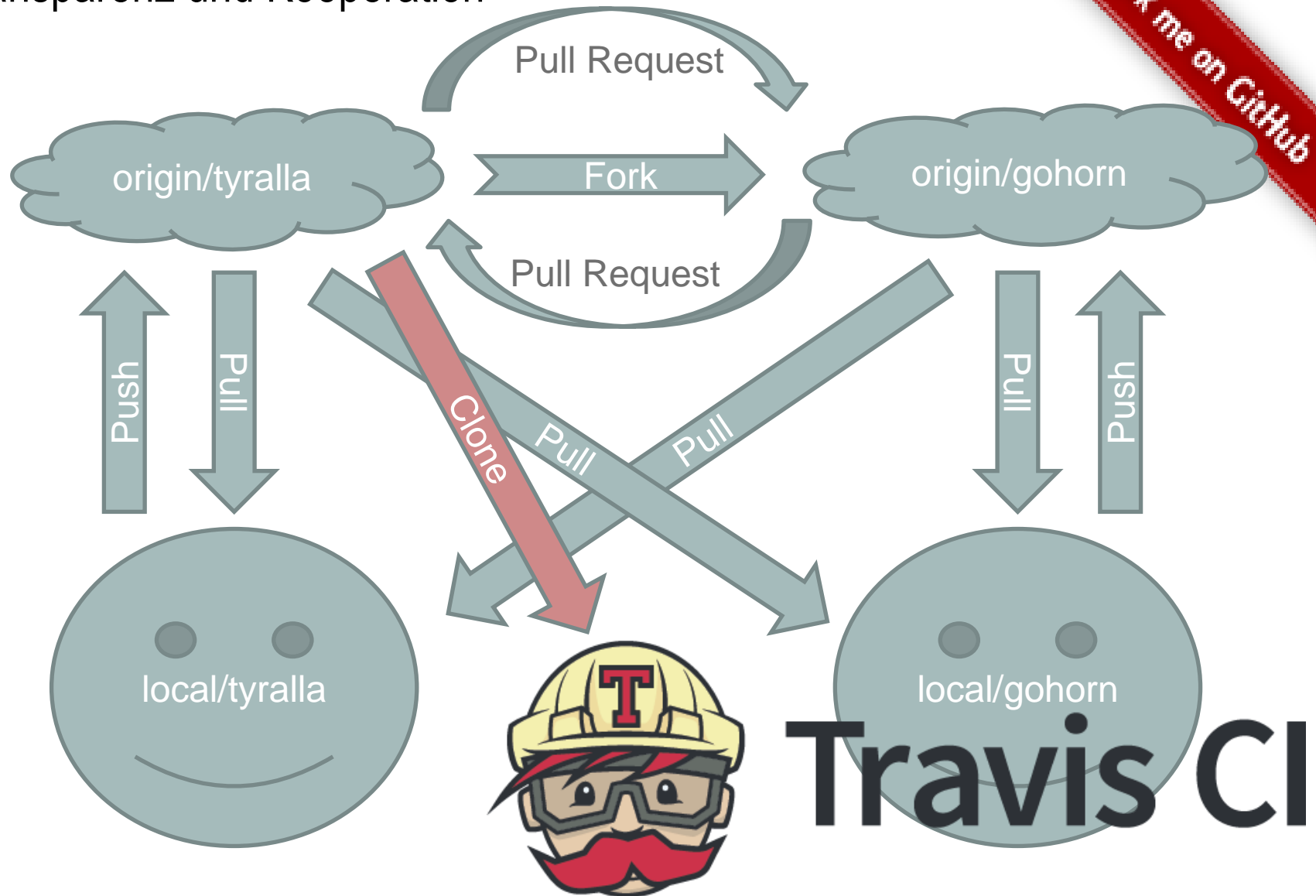
Transparenz und Kooperation

Fork me on GitHub



Transparenz und Kooperation

Fork me on GitHub



Dokumentation und Qualitätssicherung

“The only reliable documentation of a computer program is the code itself”

Brian W. Kerningham, P.J. Plauger, The Elements of Programming Style 1979

1. Integration der Dokumentation in den Quelltext
2. Integration des Quelltextes in die Dokumentation



Travis CI

Unit-Tests (Beispiel Niederschlagskorrektur)

`hydp.models.lland.lland_model.calc_nkor_v1(self)`

[\[source\]](#)

Adjust the given precipitation values.

Required control parameters:

`NHRU KG`

Required input sequence:

`Nied`

Calculated flux sequence:

`NKor`

Basic equation:

$$NKor = KG \cdot Nied$$

Example:

```
>>> from hydp.models.lland import *
>>> parameterstep('1d')
>>> nhru(3)
>>> kg(.8, 1., 1.2)
>>> inputs.nied = 10.
>>> model.calc_nkor_v1()
>>> fluxes.nkor
nkor(8.0, 10.0, 12.0)
```

Dokumentation und Qualitätssicherung

“The only reliable documentation of a computer program is the code itself”

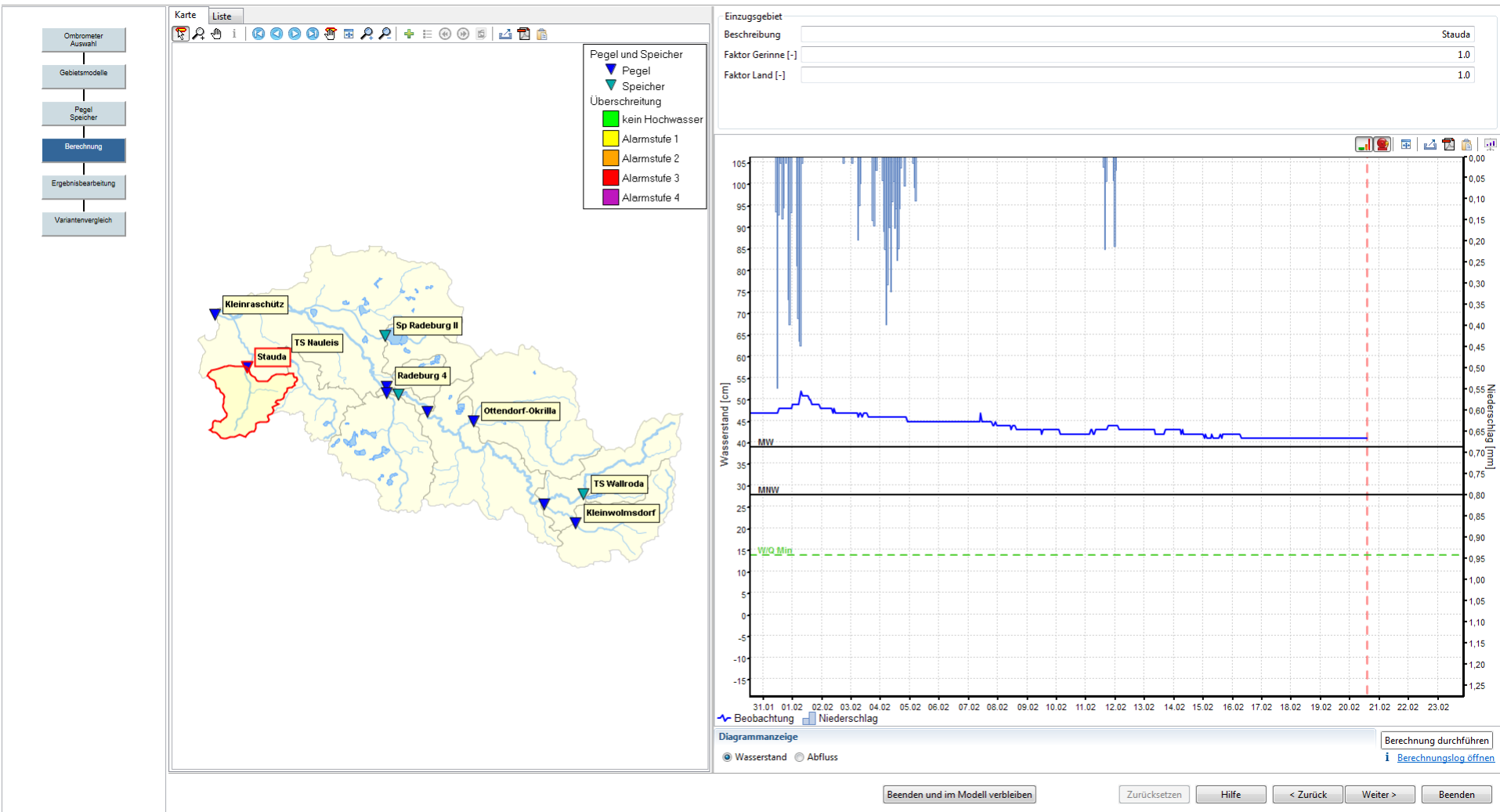
Brian W. Kerningham, P.J. Plauger, The Elements of Programming Style 1979

1. Integration der Dokumentation in den Quelltext
 2. Integration des Quelltextes in die Dokumentation
- Travis-CI:
 - installiert HydPy
 - testet HydPy auf CPython 2.7, 3.4, 3.5, 3.5dev, 3.6, 3.6dev
 - testet alle Modelle als Python und als Cython-Variante
 - Testet die Online-Dokumentation
 - Erstellt die Online-Dokumentation
 - Prüft die Testabdeckung

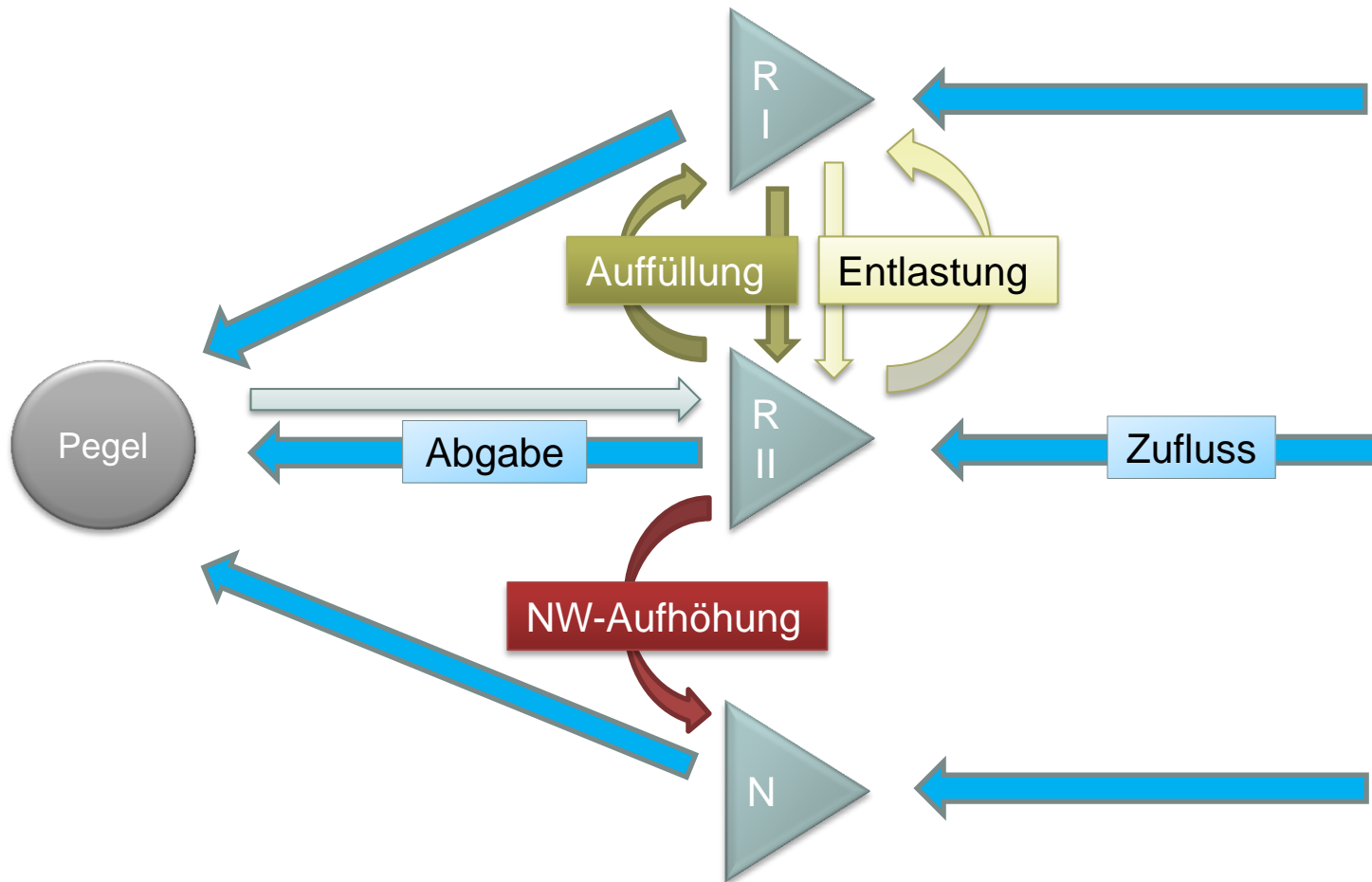


Travis CI

Integrations-Tests (Beispiel Talsperrenmodellierung)



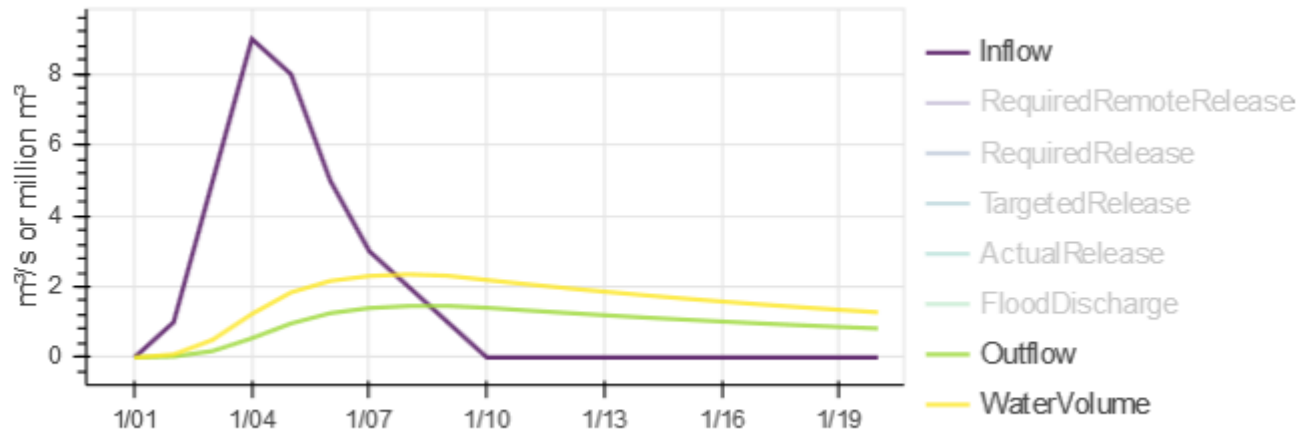
Integrations-Tests (Beispiel Talsperrenmodellierung)



Integrations-Tests (Beispiel Talsperrenmodellierung)

```
>>> test('dam_v002_ex13')
```

date	inflow	requiredremoterelease	requiredrelease	targetedrelease	actualrelease	flood
01.01.	0.0	0.0	0.0	0.0	0.0	0.0
02.01.	1.0	0.0	0.0	0.0	0.0	0.0
03.01.	5.0	0.0	0.0	0.0	0.0	0.0
04.01.	9.0	0.0	0.0	0.0	0.0	0.0
05.01.	8.0	0.0	0.0	0.0	0.0	0.0
06.01.	5.0	0.0	0.0	0.0	0.0	0.0
07.01.	3.0	0.0	0.0	0.0	0.0	0.0
08.01.	2.0	0.0	0.0	0.0	0.0	0.0
09.01.	1.0	0.0	0.0	0.0	0.0	0.0
10.01.	0.0	0.0	0.0	0.0	0.0	0.0
11.01.						
12.01.						
13.01.						
14.01.						
15.01.						
16.01.						
17.01.						
18.01.						
19.01.						
20.01.						



Laufende und anstehende Arbeiten

- Erhöhung der Testabdeckung
- Umsetzung einer statischen Quelltextanalyse
- Integration von (testbaren) Tutorials in die Online-Dokumentation
- Bereitstellung einer Executable (für Windows-Anwender)
- Elementübergreifende numerische Integration

Vielen Dank für Ihre Aufmerksamkeit!