

Open Source ENKI: Dynamic Environmental Model Framework

A possible pre-processor for WRF-Hydro?

1st European Fully Coupled Atmospheric-
Hydrological Modeling and WRF-Hydro
Users workshop. University of Calabria.
11-13 June, 2014

John F. Burkhart
Sjur Kolberg
Sigbjørn Helset
& Code
Contributors

Overview

- Energy Management in practice
- Overview of the ENKI Hydrologic Modeling Framework
- ENKI as a prototyping framework for WRF-Hydro

What is ENKI?

- Software framework for model building and evaluation
- Extension and re-writing of PINE (Rinde, 1994-97), financed by Statkraft
- Simulation of temporal processes in spatial variables
- Primary application: Distributed hydrologic models
- Now: Open Source project under operationalisation
- Sumerian deity for sweet water, wisdom and magic

The Hydrologic Modeling Framework: ENKI

- Main objects are the **model** and the **region**
- A **region** is a set of static and dynamic GIS data
- A **model** is an ordered set of subroutines
- The subroutine variables are linked to the region's GIS data.
- It's Open Source and easily available (bitbucket)
- Linux and Python Porting are underway!

The screenshot shows the Bitbucket web interface for the ENKI repository. The repository is owned by 'enkiopensource'. The page includes a navigation bar with 'Teams', 'Repositories', and 'Create' buttons. Below the repository name, there are options to 'Clone', 'Branch', and 'Pull request'. The main content area is divided into sections: 'README', 'AUTHORS', 'THANKS', 'COMING SOON', and 'COPYING / LICENSE'. The README section describes ENKI as an OpenSource hydrological toolbox and mentions its development by Statkraft. The AUTHORS section lists the principal code authors as Kolbjørn Engeland and Sjur Kolberg. The THANKS section mentions cooperation with Statkraft and the Norwegian Research Council. The COMING SOON section states that important sections will be provided as the first public release of the software is developed. The COPYRIGHT / LICENSE section indicates that ENKI is released under GPL V.3. On the right side, there is a sidebar with repository statistics: 3 Branches, 2 Tags, 0 Forks, and 4 Watchers. Below this, there is an 'Invite users to this repo' section with a 'Send invitation' button. At the bottom, there is a 'Recent activity' section showing a recent update by Sigbjørn Helset.

Bitbucket Teams Repositories Create

owner/repository

ENKI enkiopensource Share

Clone Branch Pull request

Overview Source Commits Branches Pull requests Issues 1 Wiki Downloads

README

ENKI is an OpenSource hydrological toolbox named after the ancient Sumerian god of Sweetwater, wisdom and magic.

The development of ENKI was initially financed by Statkraft for the purpose of improving hydrological forecasting for hydropower scheduling. The aim of this development is that ENKI shall become the future hydrological toolbox for hydropower companies.

For more information see: <http://www.opensource-enki.org/>

AUTHORS

ENKI has been developed at SINTEF on the basis of previous PhD works carried out at NTNU, the Norwegian University of Science and Technology. The principal code authors are Kolbjørn Engeland <Kolbjorn.Engeland@sintef.no> and Sjur Kolberg <Sjur.A.Kolberg@sintef.no>.

THANKS

Recent work has been developed in cooperation with Statkraft and other Norwegian power companies with partial complementary funding from the Norwegian Research Council.

Recent code contributors and project participants include: Sigbjørn Helset <Sigbjorn.Helset@statkraft.com> John Burkhart <John.Burkhart@statkraft.com> Oddbjørn Bruland <Oddbjorn.Bruland@statkraft.com>

COMING SOON

The following important sections will be provided as we develop the first public release of the software.

- ChangeLog
- NEWS
- INSTALL Installation instructions
- BUGS

COPYING / LICENSE

ENKI is released under GPL V.3 See the COPYING.txt file

SSH `ssh://hg@bitbucket.org/enkiopensi`

3 Branches 2 Tags 0 Forks 4 Watchers

Owner Website Access level Type Language Last updated Created Size Membership

Enki Open Source <http://www.opensourc...> Public Mercurial C++ 2014-03-26 2012-11-21 57.8 MB (download) admin (revoke)

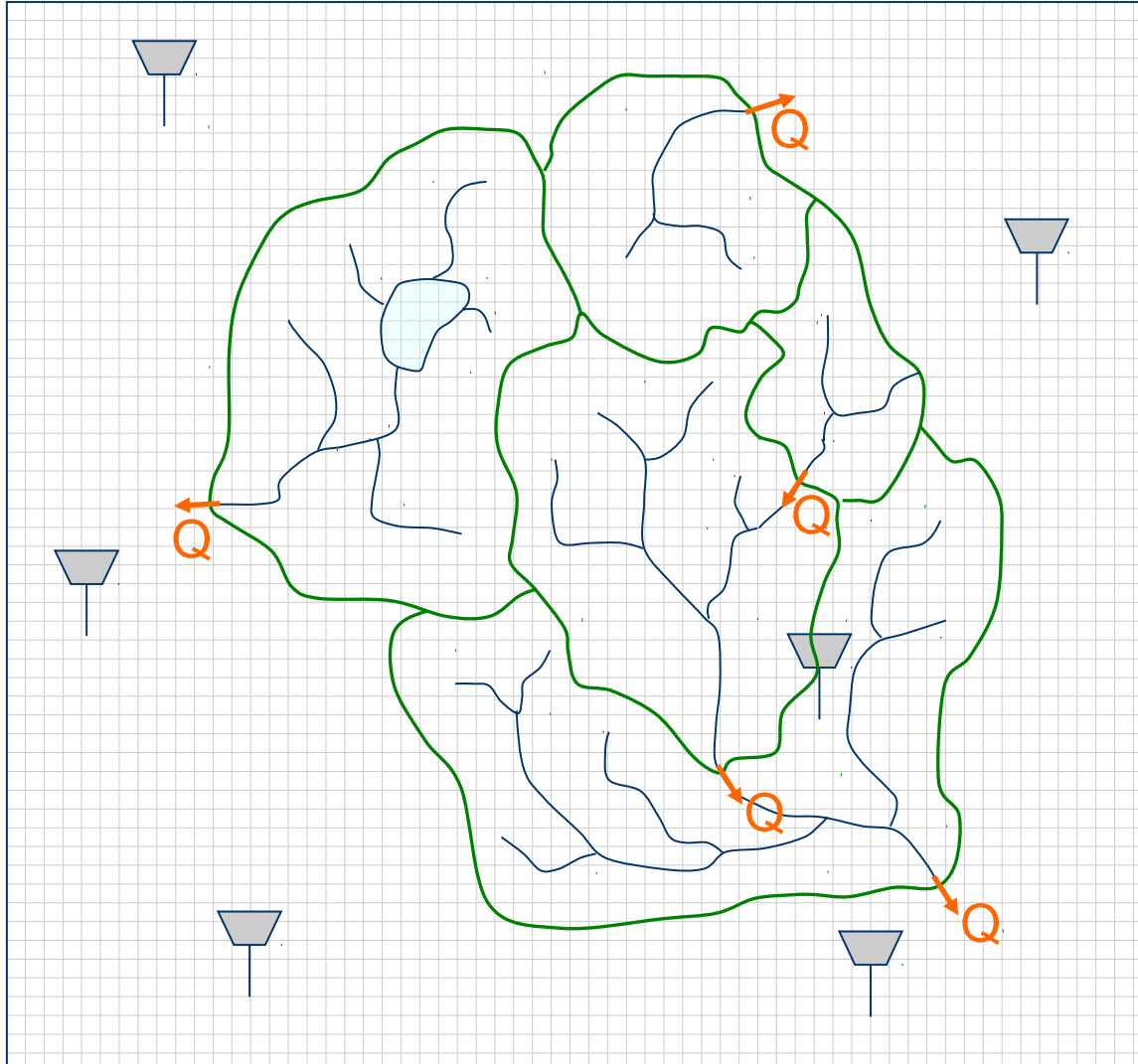
Invite users to this repo

Send invitation

Recent activity

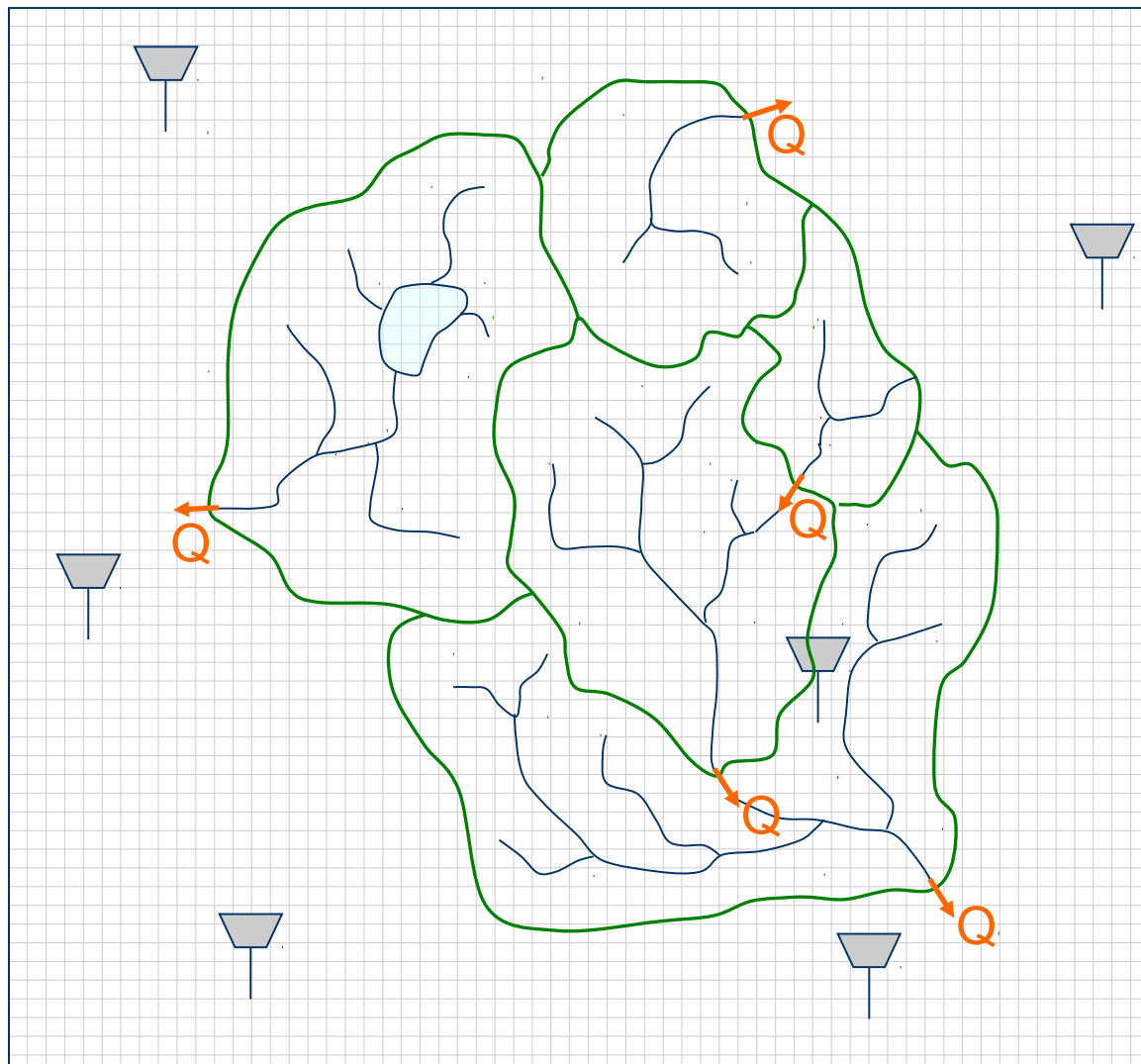
Sigbjørn Helset updated wiki-page Building Enki for Arch Linux and Ubuntu(linux)

Distributed models and regional simulation



- One regional ENKI setup replaces many HBV models
- Common setup of input data using GIS tools
- Easy input of:
 - Weather radar
 - Satellite data
 - Gridded met forecasts.
- The catchment is no longer the primary unit for
 - Simulation
 - Calibration

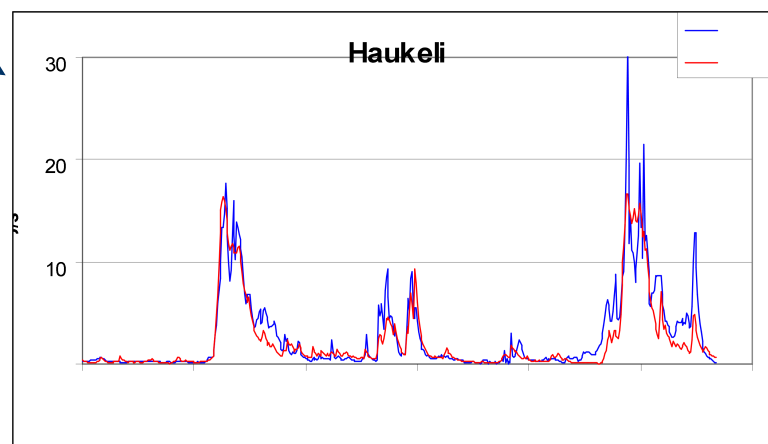
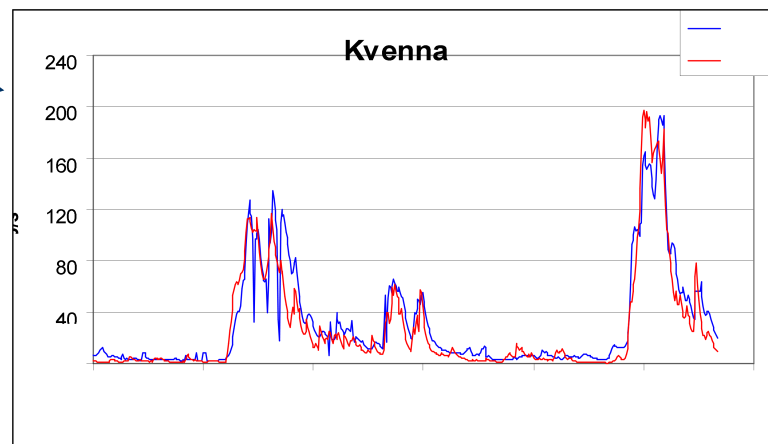
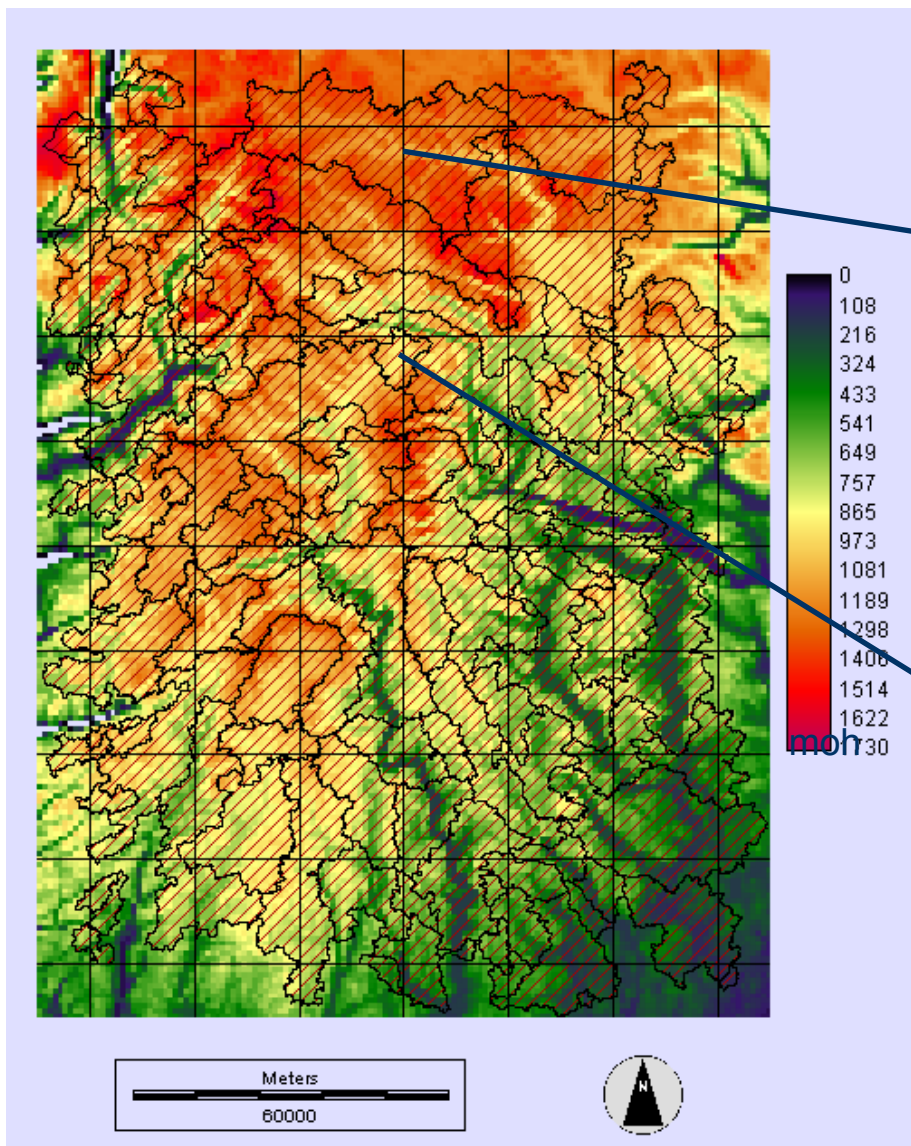
Distributed models



- Strong input gradients in mountainous basins
- Nonlinear processes
- Models need to be
 - More physical
 - Less calibrated
 - ...but simpler

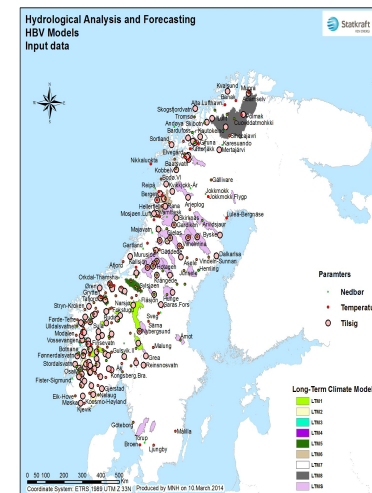
- Fewer states*
- Fewer parameters*
- Regional calibration
- Spatial parameter maps only where data allow

Regional simulation and calibration

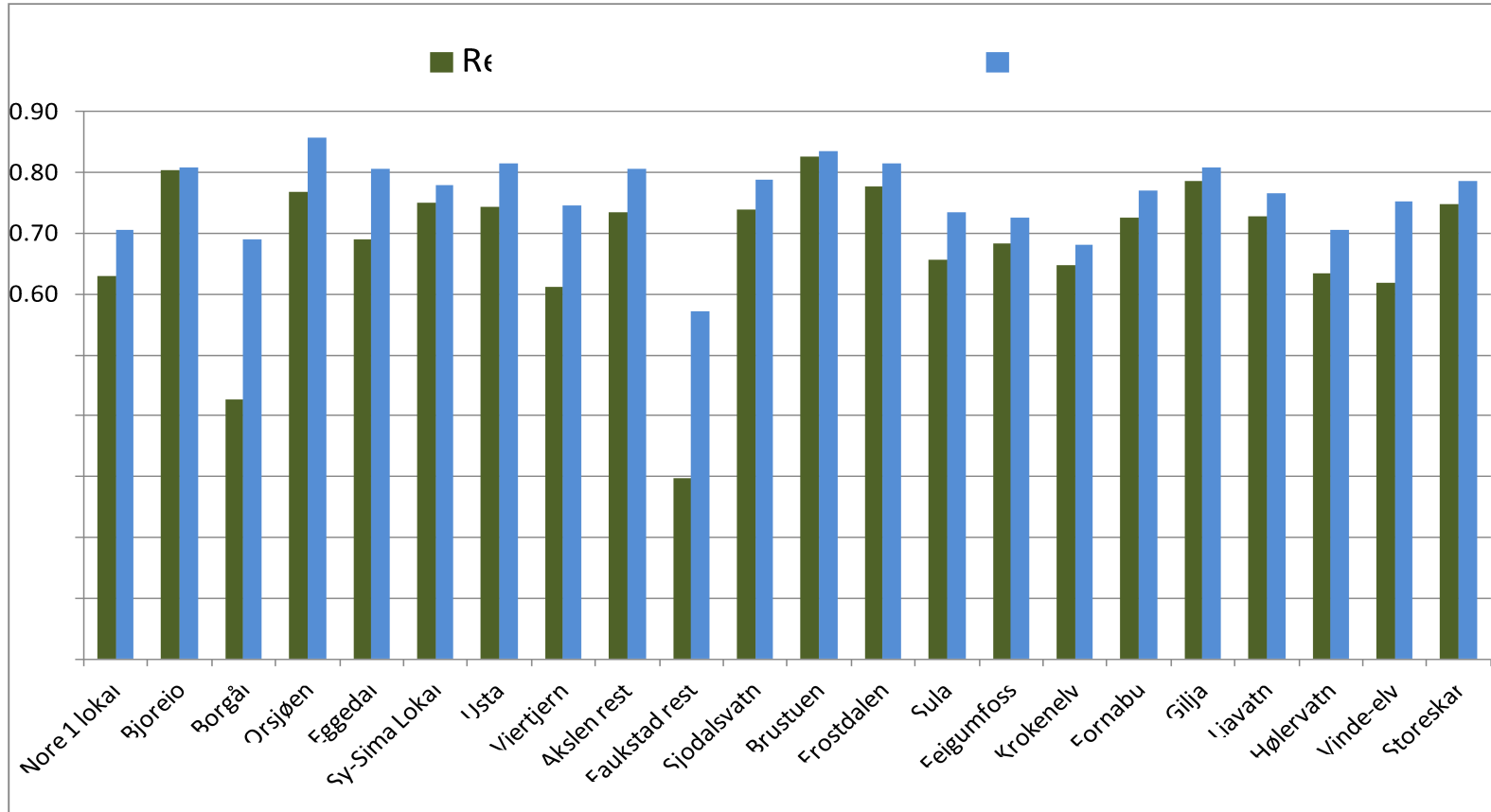


Why regional calibration?

- In a distributed model, the parameters are applied to a different spatial unit than they can be inferred from.
- Operative management: Simulate arbitrary areas / catchments
 - Compared to area scaling of discharge, differences in meteorology, elevation and land use are honoured
 - In a regional model, the performance in gauged catchments is an estimator of the performance in the ungauged catchments
- Calibration: Robust parameter estimates
 - By fitting several series simultaneously, the effect of errors and peculiarities in single catchments or series is reduced.
- Development: Discovery of errors and biases in model or data
 - Adaptation to poor observation series is avoided
 - Bad assumptions or data are exposed as errors
- **But:** Regional calibration always gives weaker performance in single catchments than specific calibration



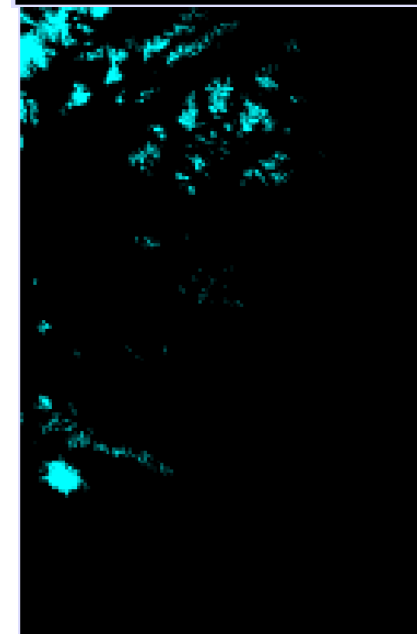
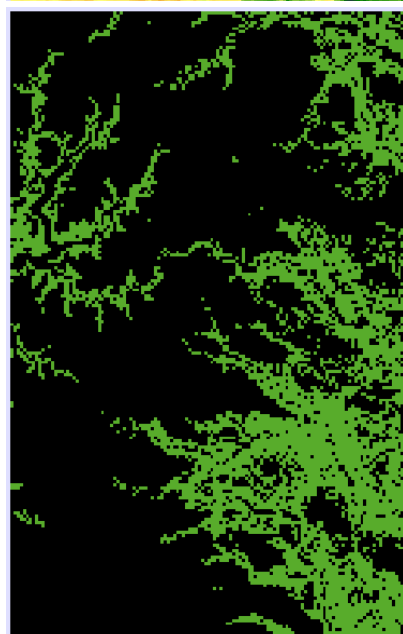
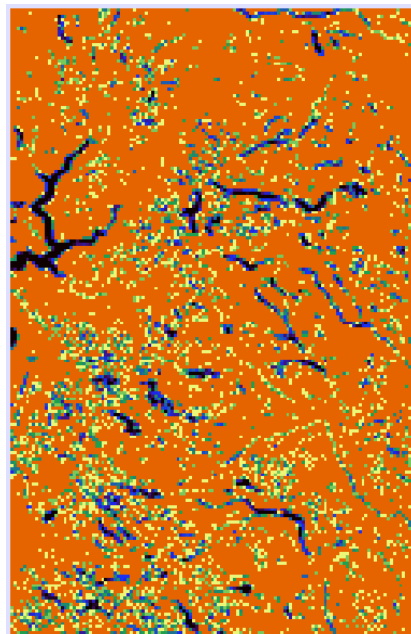
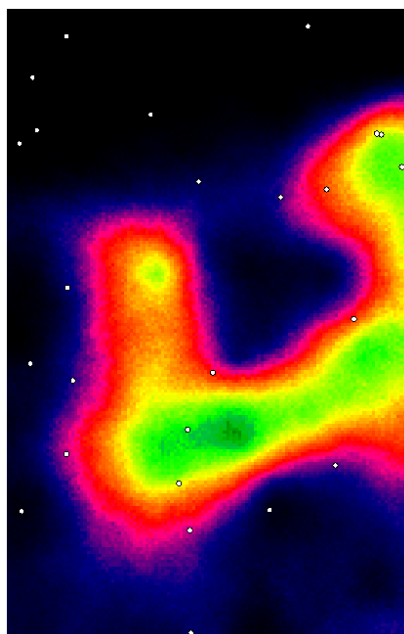
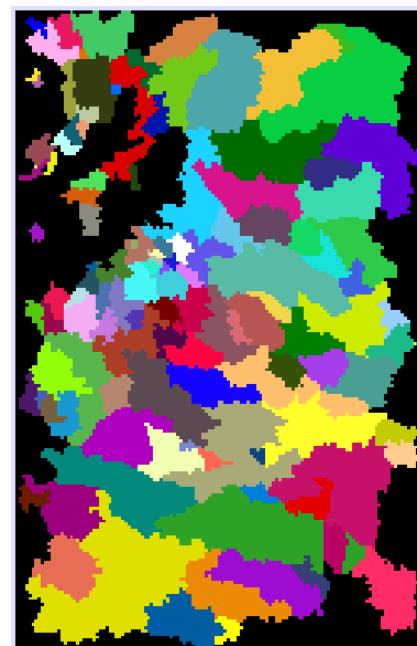
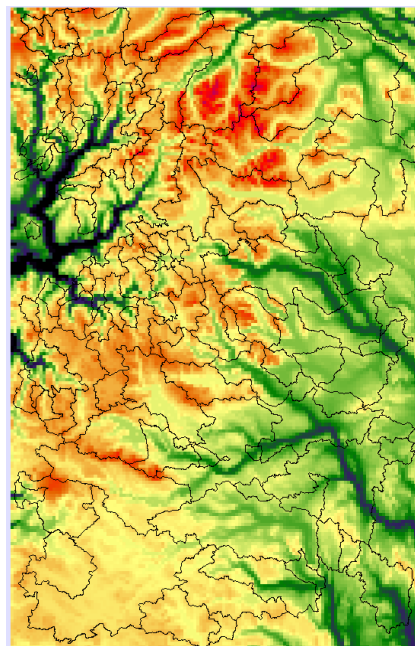
Regional and local R2



Average best R2	Regional	Local
Gridded HBV	0.679	0.741
2-par Kirchner	0.684	0.752

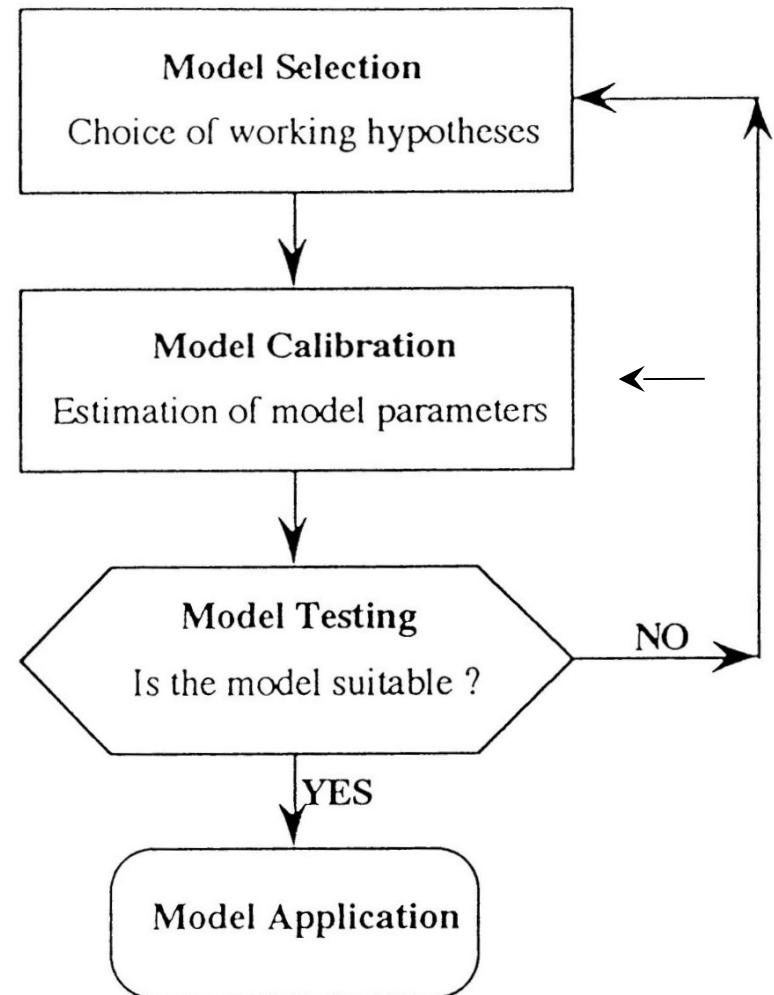
GIS based model setup:

- Elevation map (DTM)
- Subcatchment delineation
- Meteorological station map
- Lake percentage map
- Forest cover map
- Glacier cover map



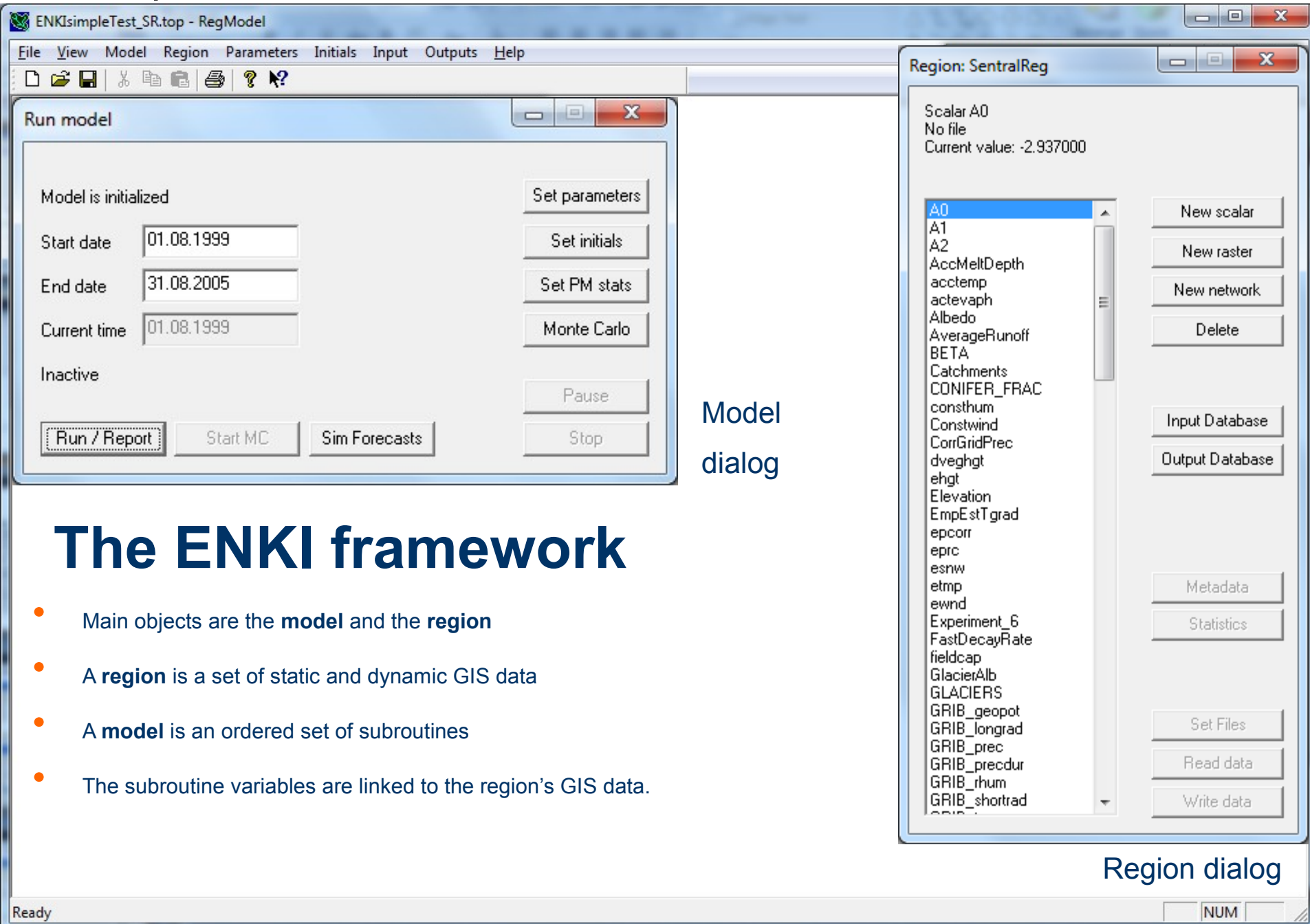
Procedure for model evaluation

- Selection:
 - Operational needs
 - Dominant processes
 - Available data
- Validation
 - Required performance
 - Achievable performance
 - Physical correctness?
- «All models are wrong. Some, however, are useful» (Box, 1979)
- «A model should not only work well, but work for the right reasons» (Klemes, 1986)



ENKI's three modes of operation

- **Model use or evaluation:** Interactive simulation, parameter estimation, import/export of GIS or TS data, construction of evaluation criteria etc.
 - No need to know the routines or internal variables in the model
- **Model building:** Link process methods to a complete model, manipulate spatial distribution
 - No need to code or compile
- **Routine development:** Implement new process routines
 - No need to handle user interface, data I/O, administration of model runs, calibration routines etc.



Model dialog

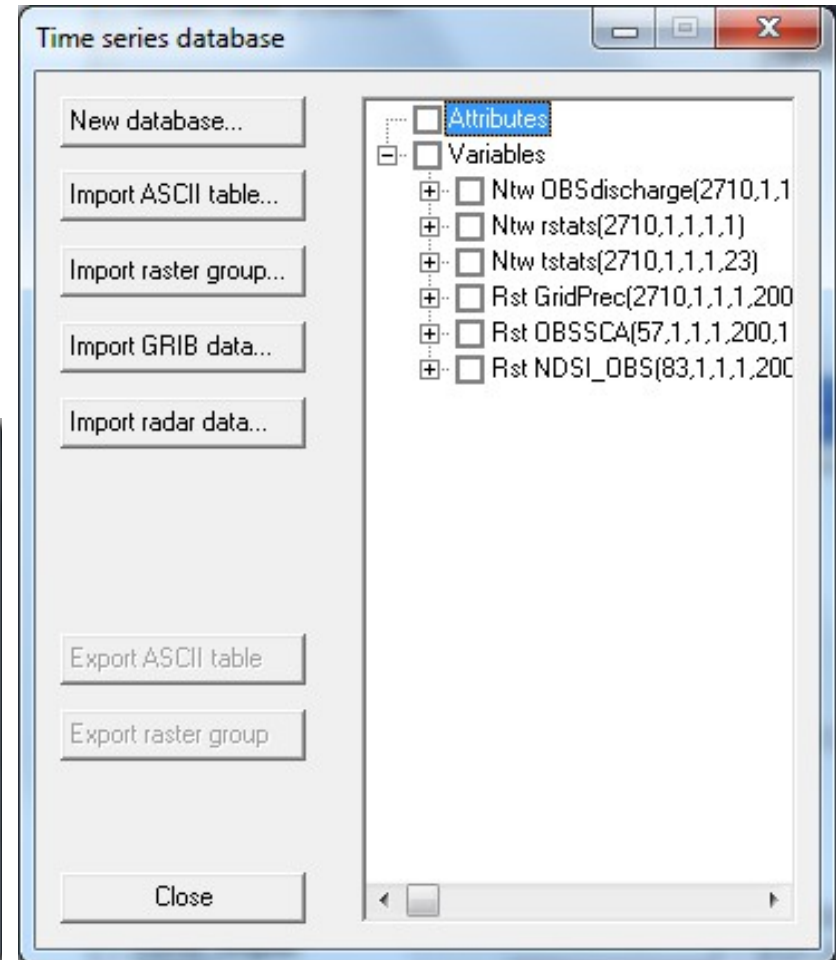
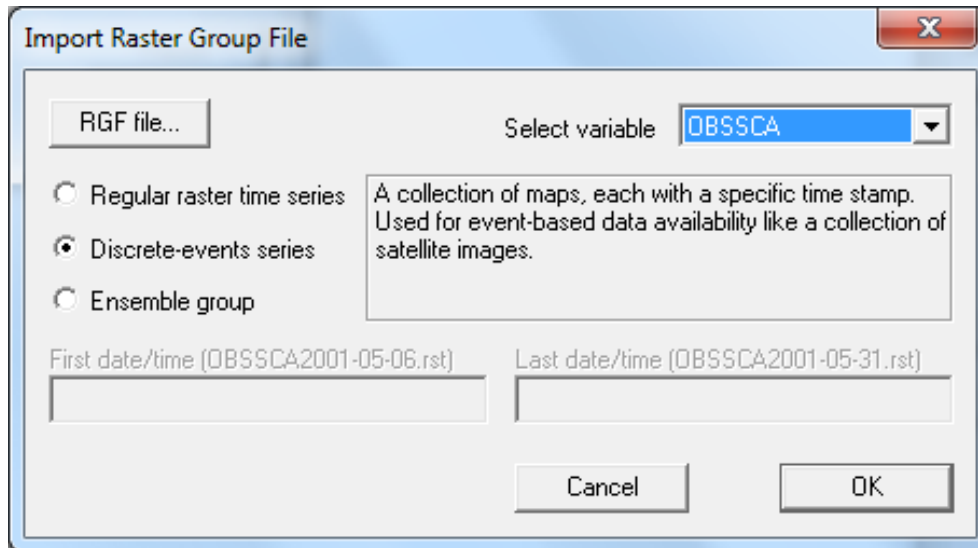
Region dialog

The ENKI framework

- Main objects are the **model** and the **region**
- A **region** is a set of static and dynamic GIS data
- A **model** is an ordered set of subroutines
- The subroutine variables are linked to the region's GIS data.

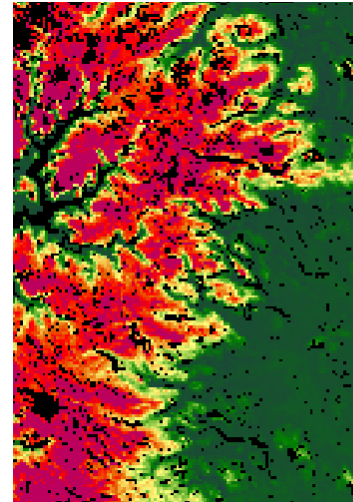
Time series import

- Time series database uses NetCDF
- Discrete time series from ASCII files
- Raster time series from map groups
- GRIB and radar file import/reproj.

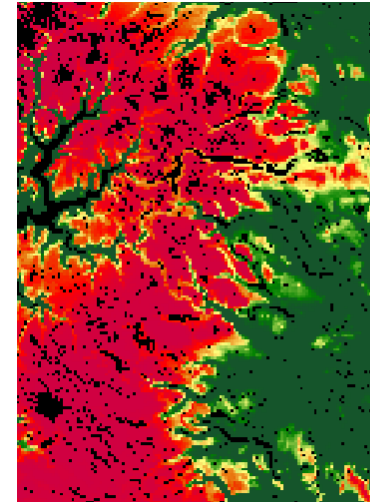


Assessing performance

- Any simulated variable is available for evaluation
- Evaluation may be temporal or spatial
- A large number of objective functions are available:
 - Additive: Information strength independent of nobs.
 - Multiplicative (likelihood): Strength increases with nobs.



Temporal evaluation



Spatial evaluation

Performance Measure specification

The image shows two overlapping windows from a software application. The background window is titled 'Performance measures specification' and contains a table with the following data:

Comparison	Test data	Reference d...	Start Time	End Time	weights
Temporal R2	SimDischarge	OBSdischarge	01.09.1999	31.08.2000	Equal
Temporal RelAvgDev	SimDischarge	OBSdischarge	01.09.1999	31.08.2000	Equal
Temporal Likelihood	SimplSimDis...	OBSdischarge	01.09.1999	31.08.2000	Equal
Spatial StdErr	SCA	OBSSCA	01.08.1999	31.08.2000	Count
Simulated values only	Upstor	none	01.08.1999	31.08.2000	None

The foreground window is titled 'New Performance Measure' and contains the following configuration options:

- Comparison type: Temporal R2
- Evaluated variable: SimDischarge
- Reference variable: OBSdischarge
- Description: Nash-Sutcliffe R2 from a simulated and a reference time series between specified dates
- Overall weighing options:
 - Equal weights
 - Data count weights
 - Average value weights
- Start time: 01.09.1999
- Stop time: 31.08.2000
- Buttons: Cancel, OK

Parameter estimation

Monte Carlo Parameter Estimation Setup

Distribution: Uniform Value: -3 Variance: 3 Min: -3 Max: 3 Set

TX (Current value: -1.05)

Parameter	Routine	Minimum	Maximum	Distribution
fieldcap	HydraSoil	0	3.40282E+0...	222.5
infcap	HydraSoil	0	3.40282E+0...	100
etmp	HydraEP	-3.40282E+...	3.40282E+0...	0.16
consthum	Consthum	-3.40282E+...	3.40282E+0...	80
BETA	HydraSoil	-3.40282E+...	3.40282E+0...	Uniform(0.5,3)
ResetSnowD...	GamSnow	0	3.40282E+0...	30
tvsum	HydraEP	0	3.40282E+0...	Uniform(100,1000)
Constwind	Constwind	-3.40282E+...	3.40282E+0...	1
TX	PcorrMap2:...	-3.40282E+...	3.40282E+0...	Uniform(-3,3)
Rtreshold	HBVRespon...	-3.40282E+...	3.40282E+0...	48
LP	HydraSoil	-3.40282E+...	3.40282E+0...	0.9
RadGrad	Idwrad	-3.40282E+...	3.40282E+0...	0
FastDecayRate	GamSnow	0	3.40282E+0...	Uniform(1,12)
Windconst	GamSnow	-3.40282E+...	3.40282E+0...	Uniform(0.5,9)
MaxIntDist	Idwrad	0	3.40282E+0...	300000
Tsill	BayesTkrig	0	3.40282E+0...	6
esnw	HydraEP	-3.40282E+...	3.40282E+0...	0.1
epcorr	HydraEP	-3.40282E+...	3.40282E+0...	1.2
Maxalbedo	GamSnow	0	1	0.9
MaxLWC	GamSnow	0	1	0.1
PriEtgrad	BayesTkrig	-3.40282E+...	3.40282E+0...	-0.6
Trange	BayesTkrig	0	3.40282E+0...	50000
Tzscale	BayesTkrig	0	3.40282E+0...	20
ewnd	HydraEP	-3.40282E+...	3.40282E+0...	0.6
perc	HBVRespon...	-3.40282E+...	3.40282E+0...	0.32
k0	HBVRespon...	-3.40282E+...	3.40282E+0...	0.007

MC method

- Marquardt-Levenberg
Multi-surface gradient search using the Jacobian matrix (PEST algorithm)
- SCE-UA
Global shuffled complex evolution. Slow and robust for difficult cases.
- Random MC (GLUE)
Random drawing from specified distributions
- DREAM MCMC
Adaptive Metropolis sampler, best used with likelihood-based PMs
- Conditional Univariate
Univariate sampling around an existing optimum, n trials per parameter dimension
- External list
Parameter sets read from file

Set Seed # MC runs: 0 Store output

Set file... 0 0 Set PM weights Cancel OK

A typical model setup

Interpolation

Interpolation

Interpolation

- The model framework does not know the function of each routine
- Each time step, the framework calls the routines in specified order
- Each routine maintains the spatial repetition loop
- The routines operate on spatial variables owned by the region
- The region copies data from/to two time series databases before and after each time step

Snow routine

Soil water / evaporation

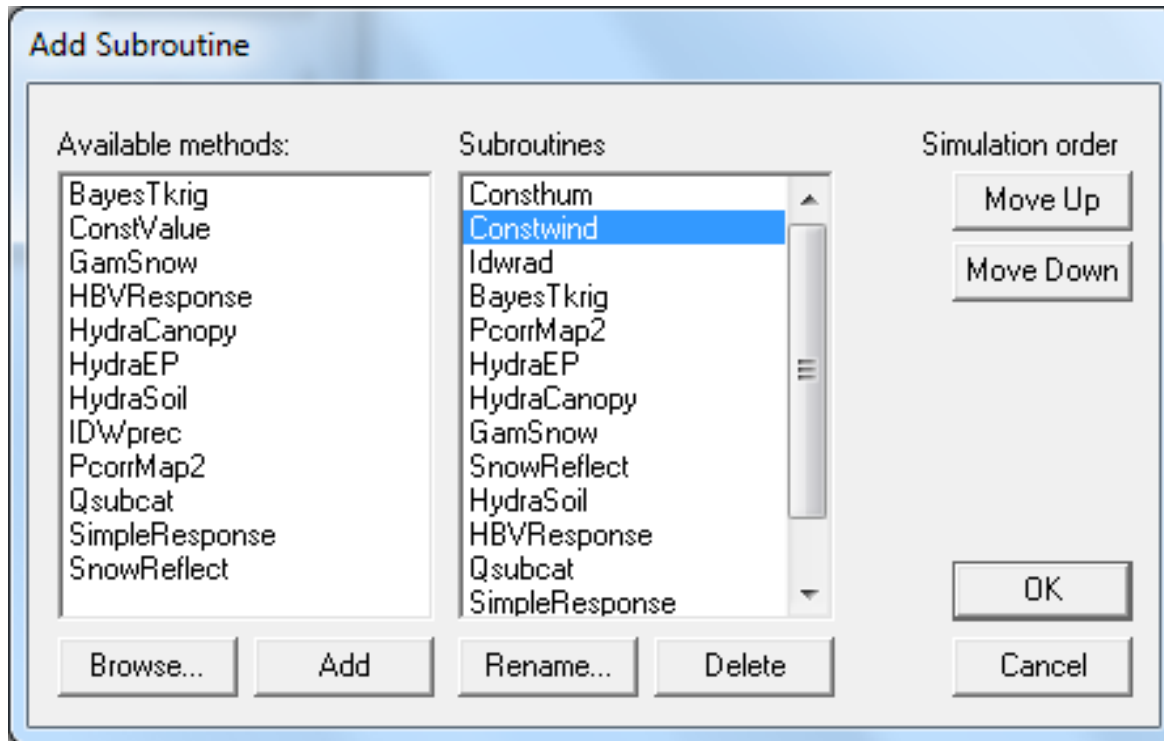
Response

Catchment aggregation

Channel routing

Results / Evaluation

Model building 1: Routine composition and order



- Loaded DLL methods in left window, with browse capability
- Selected routines in right window, where simulation order is set
- Generic methods may be used for more than one routine

Model building 2: Routine variable interface

Establish internal links in the model

HydraEP		HydraCanopy			GamSnow	
IDWtemp	HBVResponse	Qsubcat	IDWrad	ConstWind	ConstHum	HydraSoil
LocalName	Usage	Data Type	Connection	Description		
landuse	static	raster	Landuse	Delimits active area (>0) (waters=1, landsurf.>1)		
vegcov	static	raster		Coverage of high vegetation		
laicap	parameter		laicap	Storage cap. [mm] per LAI [m2]		
beta	parameter		beta	Non-linearity of soil-water retention rate		
lp	parameter		lp	Relative threshold for full transpiration		
infcap	parameter		infcap	Infiltration capacity in mm/day		
fieldcap	parameter		fieldcap	Field capacity in forested areas		
LowLAI	input		LowLAI	Leaf Area Index in non-forested areas		
HighLAI	input		HighLAI	Leaf Area Index in forested areas		
SnowOut	input		SnowOut	Water input from snow routine		
localepot	input		localepot	Local potential evaporation		
snwcov	input		snwcov	Snow coverage		
actevaph	input		actevaph	Actual evaporation from high vegetation		
solsatV	state		solsatV	Relative soil saturation in deep root zone (under fc		
solsatU	state		solsatU	Relative soil saturation in shallow root zone (no for		
llnt	state		llnt	Low vegetation interception storage		
ActEvap	state		ActEvap	Actual evaporation from all storage		

Coverage of high vegetation

vegcov Scalar Raster Network

New...
RunoffRst
Subcats
SoilOutRst

OK Cancel

Adding new variables

- Raster, point network, or scalar
- All are defined as GIS data
- Scalars have no external file
- The region "owns" the GIS variables

Add new scalar

Internal scalar identification

Scalar name: LAIcap

Flag value: -99

Data type:

Byte

Integer

Real

Point location:

Unlocated

X coord: 0

Y coord: 0

OK

Cancel

Add new raster

File: No file

File name: []

Sub-folder: Browse...

Internal raster identification:

Raster name: Vegcov

Flag value: -99

Data type:

Byte

Integer

Real

Raster geometry:

Region's default Specify

From raster File... From Region raster:

File... []

OK

Cancel

Main code modules in ENKI

RegModel.exe

Graphical user interface
Based on MFC
No visualisation, R&D oriented

Statkraft Connections

WEB based service
Links to EnkiAPI + SmG database
Daily operation oriented

EnkiAPI

Model construction
GIS and time series database
Different run functions

DHM

GIS data types
Method prototypes
GDAL integration

Methods

Method Base Class

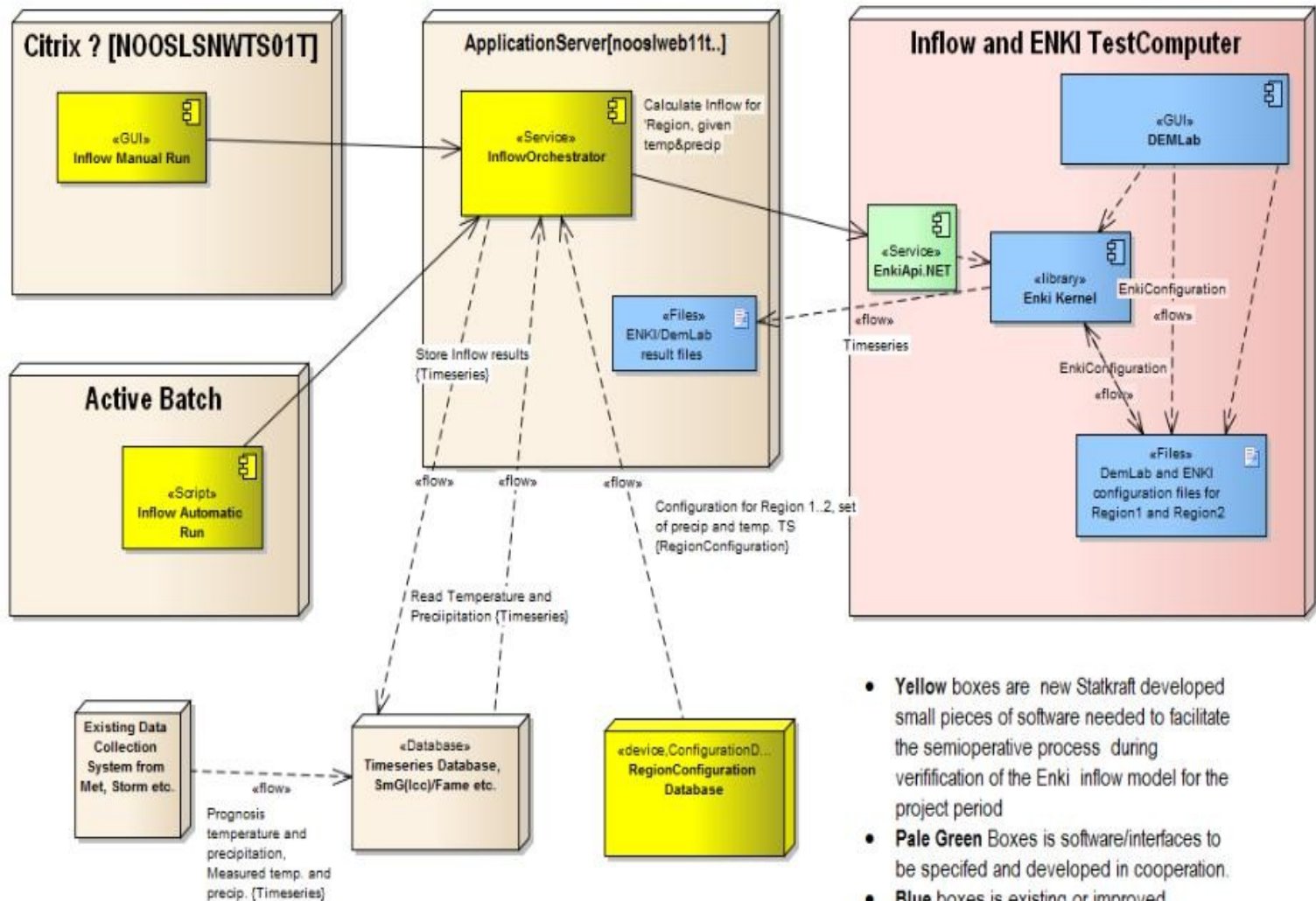
IDWtemp

HBVSoil

GamSnow

Etc

Kobling mot Statkraft-system



- **Yellow** boxes are new Statkraft developed small pieces of software needed to facilitate the semioperative process during verification of the Enki inflow model for the project period
- **Pale Green** Boxes is software/interfaces to be specified and developed in cooperation.
- **Blue** boxes is existing or improved versions of the Enki/DemLab software from Sintef.

Variable types i ENKI routines

- **Raster:** Rectangular grid, usually representing the simulation domain
- **Network:** Irregular point set, may extend outside the domain
- **Scalar:** Spatially constant quantity, located or not.

- **Static:** Time-invariant quantity not subject to calibration
 - Elevation model, lake map, forest map, gauge station map etc.
- **Parameter:** Time-invariant quantity, available for calibration
 - Can be spatially distributed (raster, network), or calibrated (scalar)
- **Input:** Dynamic variable which is read, but not written
 - Must either be in input database, or also be a response/state in earlier routine
- **Response:** Dynamic variable which is written, but not read.
 - Output variable, available for storage and evaluation
- **State:** Dynamic variable which is both read and written.
 - Need initialisation values

Available Model Methods in ENKI

- Numerous hydrologic routines have been implemented and are available for model development
- Method classes are available as part of the open source distribution
- Active development of method classes
- Documentation is poor! We are working to improve this.

Interpolation of input

Snow routines

Soil water / evaporation

Response functions

Catchment aggregation

Results / Evaluation

Automatic Calibration

Overview

- Energy Management in Statkraft
- Overview of the ENKI Hydrologic Modeling Framework
- ENKI as a prototyping framework for WRF-Hydro

ENKI's three modes of operation

- **Model use or evaluation:** Interactive simulation, parameter estimation, import/export of GIS or TS data, construction of evaluation criteria etc.
 - No need to know the routines or internal variables in the model
- **Model building:** Link process methods to a complete model, manipulate spatial distribution
 - No need to code or compile
- **Routine development:** Implement new process routines
 - No need to handle user interface, data I/O, administration of model runs, calibration routines etc.

Coding of new routines

ENKI API :: PreRun

Calls Init() for each Method

ENKI API :: RunModelBackground

MODEL :: SingleRun

PreProc()

Time Step Loop

GetInputs()

RunStep()

WriteResults()

Calls Response() of base Method Class
which in turn calls Calc()

At minimum Init() and Calc() methods
must be overloaded. If f(t) is required
Response() should be overloaded and
SetAdvancedMode() called.

Coding of new routines

ENKI API :: PreRun

Calls Init() for each Method

ENKI API :: RunModelBackground

MODEL :: SingleRun

PreProc()

Time Step Loop

GetInputs()

RunStep()

WriteResults()

Calls Response() of base Method Class
which in turn calls Calc()

At minimum Init() and Calc() methods
must be overloaded. If f(t) is required
Response() should be overloaded and
SetAdvancedMode() called.

Coding of new routines

```
#include "lineartank.h"      // LinearTank.cpp implementation

LinearTank::LinearTank()    // Constructor
{
    storconst = newmethvar("storconst",true,"methvar","parameter","Time constant, days",0);
    inflow = newmethvar("inflow",true,"methvar","input","Inflow to linear tank");
    storage = newmethvar("storage",true,"methvar","state","Response tank storage in mm",0);
    outflow = newmethvar("outflow",true,"methvar","response","Outflow from linear tank");
}

bool LinearTank::Calc()     // Response function
{
    storage->value += inflow->value;
    outflow->value = storage->value * (1 - exp(-steplength.m_span / storconst->value));
    storage->value -= outflow->value;
    return true;
}

CMethod* CreateMethodObject() // Object factory
{
    CMethod *p = new LinearTank;
    return p;
}
```

Model composition

Bayesian temperature kriging

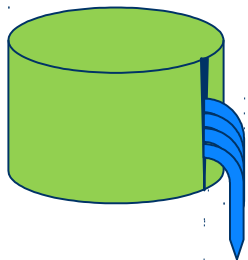
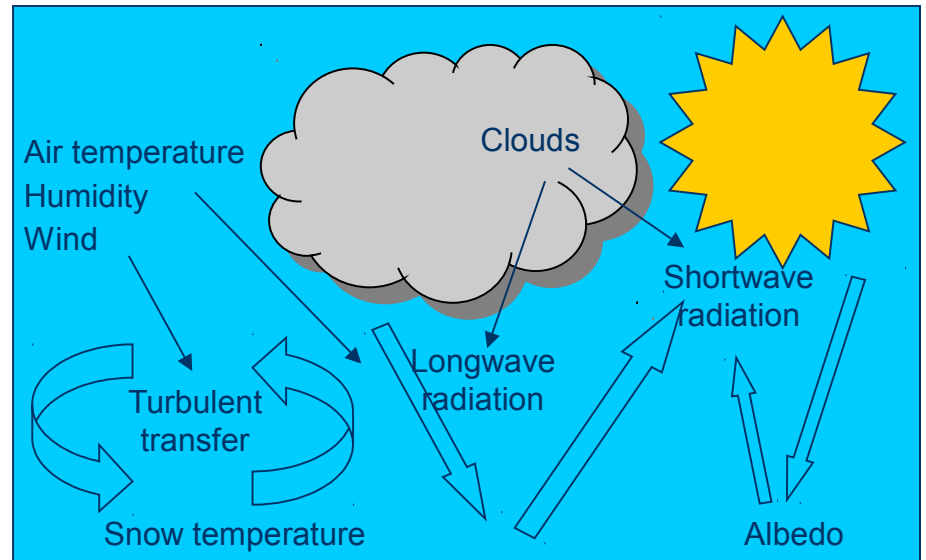
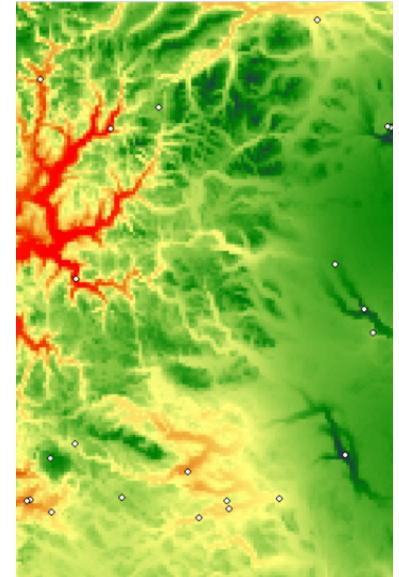
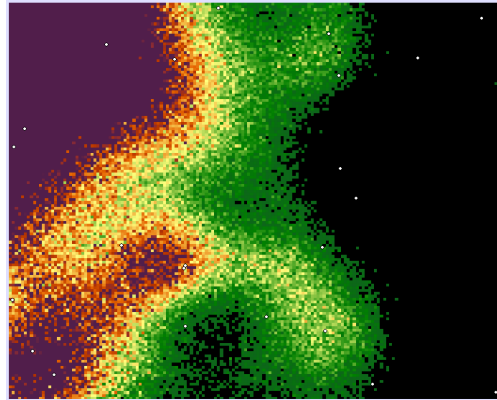
GRF precipitation simulation

Kriging of radiation, humidity, wind

Energy-sum snow melt

Priestley-Taylor potential evap

Power-law response (Kirchner, 2009)



Main code modules in ENKI

RegModel.exe

Graphical user interface
Based on MFC
Minimal visualization, R&D oriented

Statkraft Connections

WEB based service
Links to EnkiAPI + SmG database
Daily operation oriented

EnkiAPI

Model construction
GIS and time series database
Different run functions

DHM

GIS data types
Method prototypes
GDAL integration

Template Methods

IDWtemp

GamSnow

Method Base Class

HBVSoil

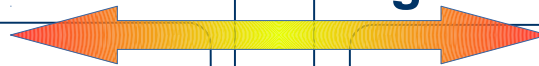
Python Algorithms

IDWtemp

GamSnow

Method Base Class

HBVSoil

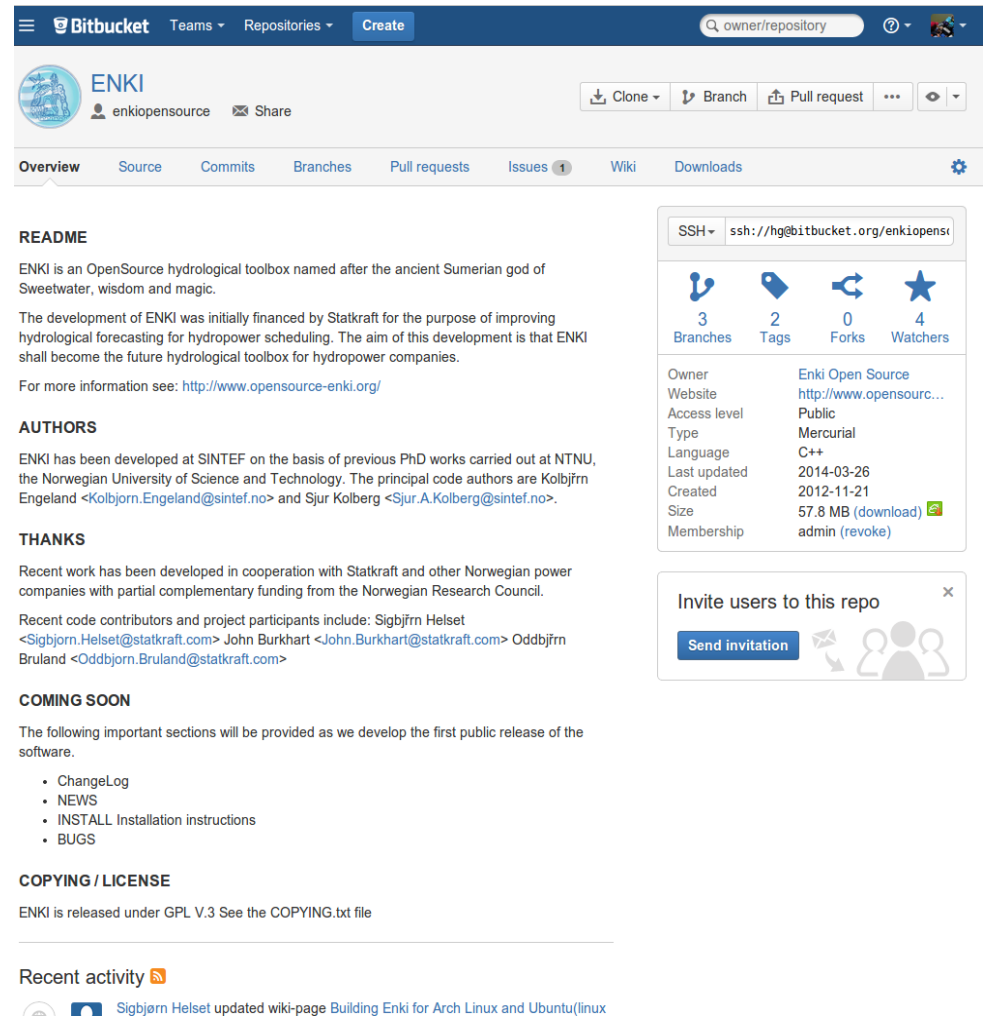


Summary

- ENKI an excellent tool for testing different model compositions and spatial setup alternatives
- Distributed modelling requires re-thinking of many subroutines, in particular regarding input data
- Regional calibration and restriction of calibration freedom show modest reduction of performance
- Improved parameter identifiability enables regionalisation
- ENKI recently released as OS under the LGPL license
- Currently in operationalisation phase both at Statkraft and among other companies

The Hydrologic Modeling Framework: ENKI

- Search: “ENKI Hydrology”
- Bitbucket is source for linux branch



The screenshot shows the Bitbucket web interface for the ENKI repository. The repository is owned by 'enkiopensource' and is public. The main content area displays the README, which describes ENKI as an OpenSource hydrological toolbox. It mentions that the development was initially financed by Statkraft for improving hydrological forecasting for hydropower scheduling. The README also lists authors (Kolbjørn Engeland and Sjur Kolberg) and thanks contributors (Sigbjørn Helset, John Burkhart, and Oddbjørn Bruland). A 'COMING SOON' section indicates that important sections will be provided as the first public release of the software is developed. The 'COPYING / LICENSE' section states that ENKI is released under GPL V.3. On the right side, there are statistics for the repository: 3 branches, 2 tags, 0 forks, and 4 watchers. There is also an 'Invite users to this repo' button and a 'Recent activity' section at the bottom showing a recent update by Sigbjørn Helset.

Bitbucket Teams ▾ Repositories ▾ Create

owner/repository

ENKI
enkiopensource Share

Clone ▾ Branch Pull request ⋮

Overview Source Commits Branches Pull requests Issues 1 Wiki Downloads

README

ENKI is an OpenSource hydrological toolbox named after the ancient Sumerian god of Sweetwater, wisdom and magic.

The development of ENKI was initially financed by Statkraft for the purpose of improving hydrological forecasting for hydropower scheduling. The aim of this development is that ENKI shall become the future hydrological toolbox for hydropower companies.

For more information see: <http://www.opensource-enki.org/>

AUTHORS

ENKI has been developed at SINTEF on the basis of previous PhD works carried out at NTNU, the Norwegian University of Science and Technology. The principal code authors are Kolbjørn Engeland <Kolbjorn.Engeland@sintef.no> and Sjur Kolberg <Sjur.A.Kolberg@sintef.no>.

THANKS

Recent work has been developed in cooperation with Statkraft and other Norwegian power companies with partial complementary funding from the Norwegian Research Council.

Recent code contributors and project participants include: Sigbjørn Helset <Sigbjorn.Helset@statkraft.com> John Burkhart <John.Burkhart@statkraft.com> Oddbjørn Bruland <Oddbjorn.Bruland@statkraft.com>

COMING SOON

The following important sections will be provided as we develop the first public release of the software.

- ChangeLog
- NEWS
- INSTALL Installation instructions
- BUGS

COPYING / LICENSE

ENKI is released under GPL V.3 See the COPYING.txt file

SSH `ssh://hg@bitbucket.org/enkiopensi`

3 Branches 2 Tags 0 Forks 4 Watchers

Owner	Enki Open Source
Website	http://www.opensourc...
Access level	Public
Type	Mercurial
Language	C++
Last updated	2014-03-26
Created	2012-11-21
Size	57.8 MB (download)
Membership	admin (revoke)

Invite users to this repo

Send invitation

Recent activity

Sigbjørn Helset updated wiki-page Building Enki for Arch Linux and Ubuntu(linux)

Thank you all.

**John F. Burkhart
Sjur Kolberg
Sigbjørn Helset
& Code
Contributors**