JGrasstools

Andrea Antonello
Silvia Franceschi
Riccardo Rigon
Maurizio Righetti

19 Novembre 2010

SEMPRE E COMUNQUE STANDARD

Storia: il team di JGrass migra tutti i propri modelli allo standard di modellazione OpenMI 1.4 (Open Modelling Interfaces). La migrazione finisce nel 2009.

Nello stesso anno si comincia a lavorare a OpenMI 2.0. Le interfacce di sviluppo cambiano **NON** poco. All'orizzonte c'e' un porting da 1.4 a 2.0 di qualcosa come 60 moduli.

SEMPRE E COMUNQUE STANDARD

Inoltre l'esperienza OpenMI ci mostra che:

- l'associazione OpenMI tiene aggiornata la versione .NET, meno quella Java
- Il framework di sviluppo OpenMI e' molto invasivo. Richiede di adattare i moduli a una serie di interfacce
- Gli altri partner OpenMI basano su software chiuso (Deltares-Delft Hydraulics, DHI, Wallinford...). Questo non solo non ha alcun vantaggio per JGrass, ma rende impossibile il testing dei modelli collegati ad altri software

SEMPRE E COMUNQUE STANDARD

Si cercano possibili soluzioni e si trova:
Object Modelling System [0] - OMS3

- E' un framework di modellazione a componenti
- Utilizza annotazioni sul codice invece di imporre interfacce e tipi di dati
- Nasce come progetto per la creazione di una libreria di modelli ambientali FOSS [1]
- Viene sviluppato dal dipartimento dell'agricoltura americano e la Colorado State University

Si comincia a collaborare sul progetto di migrazione, l'USDA mette a disposizione lo sviluppatore principale per aiutare nella prima fase.

[0] http://www.javaforge.com/project/oms
[1] http://www.javaforge.com/proj/tracker/browseTracker.do?tracker_id=7040

IMPARARE DAL PASSATO

Si inizia la migrazione a OMS3, ma imparando dal passato si decide di:

- Non applicare tale migrazione direttamente a JGrass, onde evitare di destabilizzarlo
- Creare una libreria, in modo da rendere gli strumenti disponibili anche per processing fatto via web
- Fare pulizia nel codice, fortemente penalizzato dalla “gabbia“ OpenMI
- Tornare alla licenza GPL che ci e' in generale piu' simpatica

IMPARARE DAL PASSATO

Creare delle regole ben definite per i contributi. Un modulo puo' entrare in una release solo se:

1) Il codice segue alcune regole della casa
2) Il modulo ha un testcase associato
3) Il modulo ha una pagina di documentazione nel WIKI (che deve contenere una descrizione, un collegamento ad un esempio di utilizzo e dove possibile un'immagine)
4) ...avere uno script di esempio?

GEOSCRIPTING

I jgrasstools possono essere utilizzati anche in ambiente di scripting.

Il linguaggio di scripting utilizzato come collante e' Groovy [0].

Il linguaggio per la modellazione ambientale, il collegamento fra modelli e la loro esecuzione e' quello definito da OMS3.

Per il linguaggio di scripting GIS si sta lavorando all'integrazione di geoscript [1].

[0] http://groovy.codehaus.org
[1] http://geoscript.org/

CONTENUTI

I jgrasstools, seguendo il percorso iniziato con JGrass, sono dedicati all'analisi geomorfologica e all'idrologia.

CONTENUTI

jgrasstools
The geospatial toolbox that backs JGrass

Project Home | Wiki | Issues | Source | Administer

New page | Search Current pages for Search | Edit this page | Delete this page

- The Horton Machine
- Gears
  - I/O
  - Raster modules
    - CoverageConverter
    - CoverageReprojector
    - CutOut
    - Mapcalc
    - ScanLineRasterizer
    - CoverageSummary
  - Vector modules
    - AttributesJoiner
    - AttributesRounder ?
    - FeatureFilter ?
    - FeatureReprojector
    - FeatureReshaper ?
    - GeometrySimplifier
    - LineSmoother
    - MarchingSquaresVectorializer
  - Modules Overview
- OMS3
- Developer documentation

Edit

Mapcalc

The raster map algebra module

Description

The module takes a set of rasters on which a supplied equation can be applied on a pixel by pixel basis. The module uses the jiffle library for the calculations. For a deeper look into its features, refer to its website. For an example inside the jgrasstools see below.

Example usage

An example usage of the algorithm can be found in the testcases suite: Mapcalc

Example map

Strumenti di lettura e scrittura e strumenti di accesso ai dati, modifica, rasterizzazione, riproiezione e simili.

CONTENUTI

jgrasstools
The geospatial toolbox that backs JGrass

Project Home | Wiki | Issues | Source | Administer

New page | Search Current pages for Search | Edit this page | Delete this page

- The Horton Machine (Edit)
  - Dem Manipulation
    - Pitfiller
    - Wateroutlet
  - Geomorphology
    - Ab
    - Aspect
    - Curvatures
    - DrainDir
    - FlowDirections
    - Gradient
    - Slope
    - Tca
    - Tca3D
  - Network Related Measures
  - Hillslope Analyses
  - Basin Attributes
    - RescaledDistance
    - TopIndex
    - BasinShape
  - Statistics
    - Cb
    - Jami
    - Kriging
  - Hydrogeomorphology
- Gears
- Modules Overview
- OMS3
- Developer documentation

RescaledDistance

The rescaled distances module

Description

Calculates the rescaled distance of each pixel from the outlet. Such distance is so defined:

$x = x_c + rx_h$

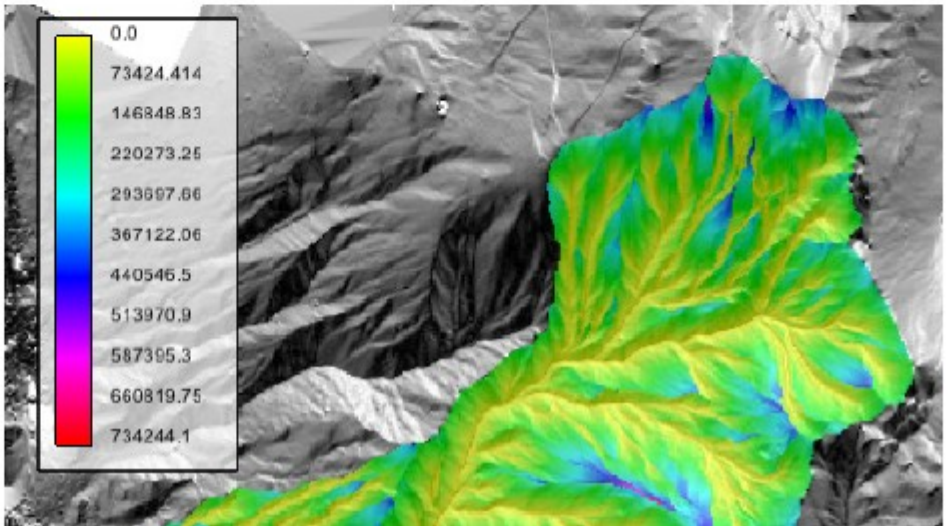
where:

- x_c is the distance along the channels,
- $r = {}^{c}/_{ch}$ the ratio between the speed in the channel state, c and the speed in the hillslopes, ch
- x_h the distance along the hillslopes.

Example usage

An example usage of the algorithm can be found in the testcases suite: RescaledDistance

Example map



2) Moduli abbastanza standard di analisi geomorfologica e di bacino

CONTENUTI

jgrasstools
The geospatial toolbox that backs JGrass

Project Home | Wiki | Issues | Source | Administer

New page | Search Current pages for Search | Edit this page | Delete this page

- The Horton Machine (Edit)
  - Dem Manipulation
  - Geomorphology
  - Network Related Measures
  - Hillslope Analyses
  - Basin Attributes
  - Statistics
  - Hydrogeomorphology
    - Adige
    - EnergyBalance
    - EnergyIndexCalculator
    - Peakflow
    - Shalstab
    - Insolation
    - Hillshade
    - Sky View Factor
  - Gears
  - Modules Overview
  - OMS3
  - Developer documentation

EnergyBalance

The snow model

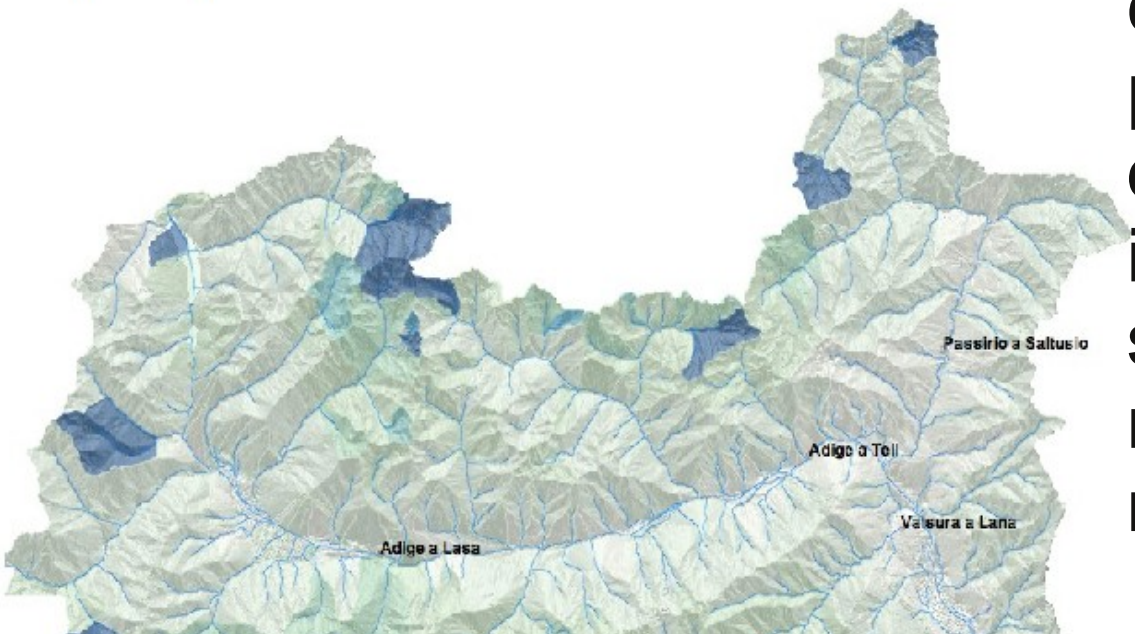
Description

- solves the energy balance for each hillslope dividing it in altimetric and energetic bands
- contains a sophisticated snow module with which it is possible to calculate the SWE for all the basins
- evaluation of the contribution to the discharge from the glaciers during summer time in term of net precipitation (P + SM - SA)
- results are presented in term of average in watershed

Example usage

An example usage of the algorithm can be found in the testcases suite: EnergyBalance

Example map



Modelli complessi per il calcolo del ciclo idrologico, scioglimento nivale e radiazione

ESEMPI DI APPLICAZIONE: COMMANDLINE

La compilazione dei jgrasstools genera anche un pacchetto dei jgrasstools che contiene tutte le librerie necessarie per essere usato in modalita' standalone.

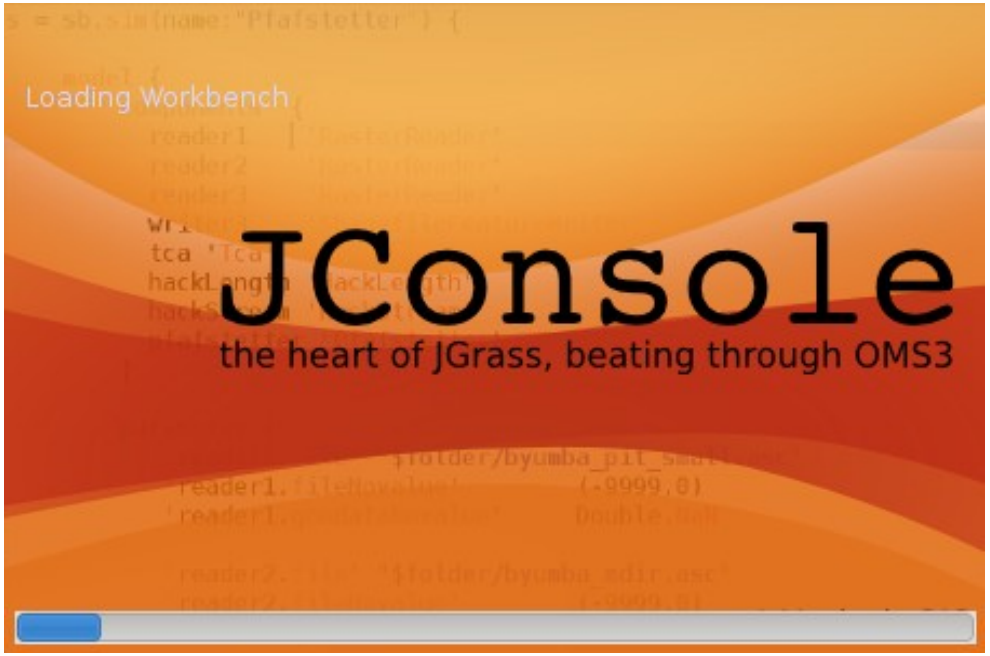
Da riga di comando si ottiene:

~\development\jgrasstools-hg\jgrasstools\hortonmachine\target >>> **java -jar hortonmachine-all.jar**
USAGE:
java -jar jgrasstools.jar
PathToScript → **percorso completo allo script da eseguire**
[--work <working_folder_path>] → **eventuale percorso da usare come working folder**
[--mode <loglevel> → **il livello di logging per l;identificazione degli errori**

ESEMPI DI APPLICAZIONE: JConsole

JConsole e' un ambiente di supporto allo scripting con OMS3 e jgrasstools.

La utilizziamo per usare i jgrasstools fino a quando non sara' il momento di inserire i jgrasstools come motore dentro a JGrass/uDig.



ESEMPI DI APPLICAZIONE: JConsole

Completamento dei comandi e colorazione della sintassi

Customizzazione dei processi, memoria dedicata, livello di logging

JConsole

File Edit Help

PitFiller.jgrass

```
def pit = "dtm_test_pit.asc"

s = sb.sim(name:"Pitfiller") {

    model {
        components  {
          reader    'RasterReader'
          writer    'ArcgridCoverageWriter'
          pitfiller 'Pitfiller'
        }

        parameter {
           'reader.file' "$folder/$dtm"
           'writer.file' "$folder/$pit"
           // optional setting of novalues, in this case same as default
           //'reader.fileNovalue'       -9999.0
           //'reader.geodataNovalue'    Double.NaN
           'pitfiller.pm' pm
        }

        connect {
          'reader.geodata' 'pitfiller.inDem'
          'pitfiller.outPit' 'writer.g
```

inDem
pm
outPit
doProcess

Console

JConsole Output

```
Initialize boundaries...
Initialize internal pointers...
Set positive slope directions...
Number of pits to resolve: 3
Main cycle...
Approximate estimate of work done:
0%
Pitfiller finished...
Process finished: 2010-07-04 12:19:12
```

Log level OFF | Process memory [MB] 512 | Writable | Insert

ESEMPI DI APPLICAZIONE: JConsole

Editing propri di un ambiente di siluppo

JConsole

File Edit Help

Undo Ctrl+Z
Redo Ctrl+Y
Cut Ctrl+X
Copy Ctrl+C
Paste Ctrl+V
Find Incremental Ctrl+J
Find/Replace Ctrl+F
Find Next Ctrl+K
Find Previous Shift+Ctrl+K
Goto Line Ctrl+L
Show Whitespaces
Word Completion Alt+/
Command Completion Ctrl+Space

```
ter'
ues, in this case same as default
        //'reader.fileNovalue'        -9999.0
        //'reader.geodataNovalue'     Double.NaN
        'pitfiller.pm' pm
    }

    connect {
      'reader.geodata' 'pitfiller.inDem'
      'pitfiller.outPit' 'writer.geodata'
```

Console

JConsole Output

```
Initialize boundaries...
Initialize internal pointers...
Set positive slope directions...
Number of pits to resolve: 3
Main cycle...
Approximate estimate of work done:
0%
Pitfiller finished...
Process finished: 2010-07-04 12:19:12
```

Log level OFF | Process memory [MB] 512 | Writable | Insert

ESEMPI DI APPLICAZIONE: JConsole

Processi multipli

JConsole

File Edit Help

FlowDirection.jgrass | NetNumbering.jgrass | NetworkExtraction... | Pfafstetter.jgrass

```
s = sb.sim(name:"Pfafstetter") {

    // your folder here
    def folder = new File("/home/silli/lavori/jgrasstools").absolutePath;

    model {
        components  {
          reader1    'RasterReader'
          reader2    'RasterReader'
          reader3    'RasterReader'
          writer3    'ShapefileFeatureWriter'
          tca 'Tca'
          hackLength 'HackLength'
          hackStream 'HackStream'
          pfafstetter 'Pfafstetter'
        }

        parameter {
           'reader1.file' "$folder/byumba_pit_small.asc"
           'reader1.fileNovalue'        (-9999.0)
           'reader1.geodataNovalue'     Double.NaN

           'reader2.file' "$folder/byumba_mdir.asc"
```

Console

JConsole Output

```
Process started: 2010-07-04 12:22:32
```

Log level OFF | Process memory [MB] 512 | Writable | Insert

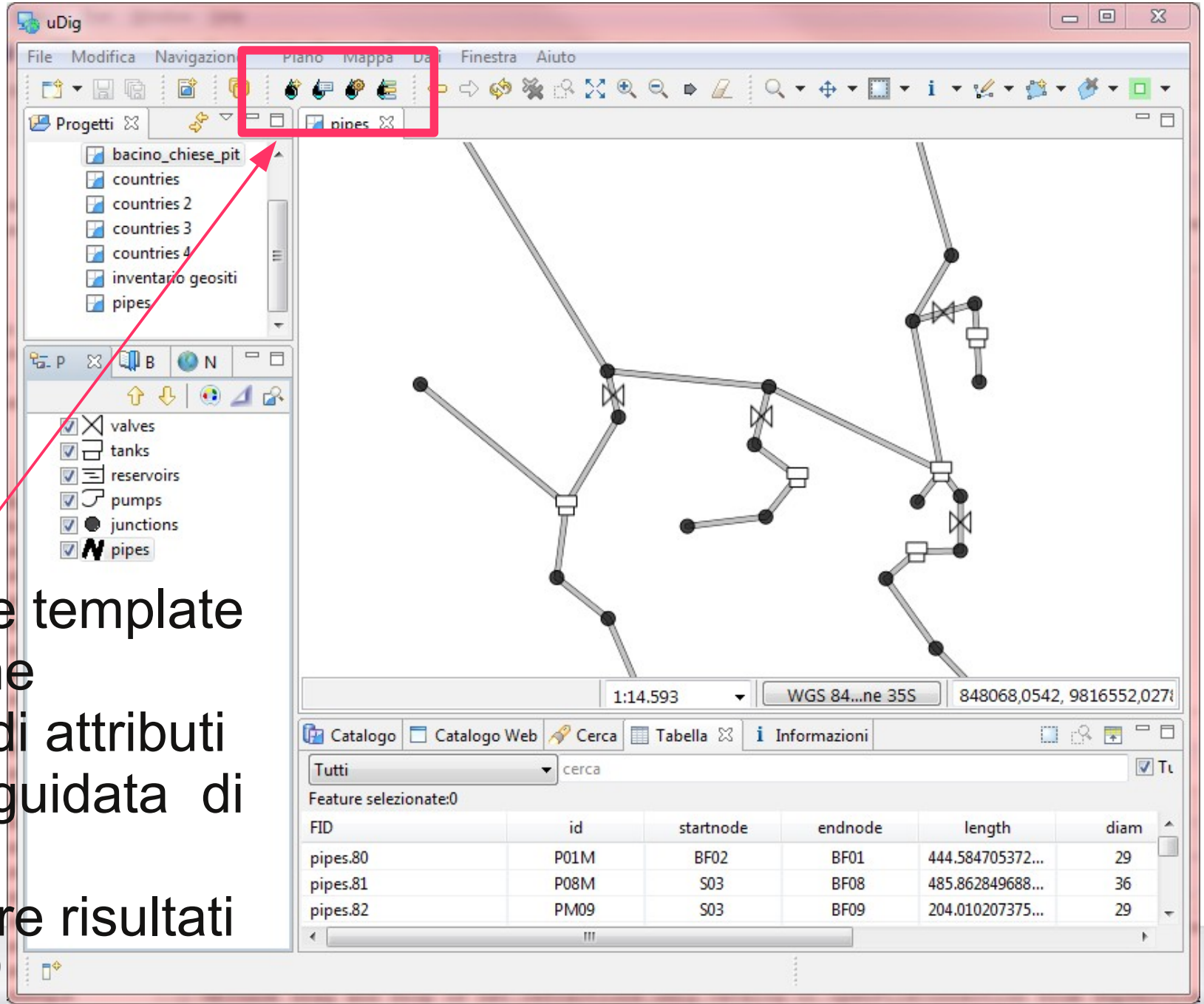
ESEMPI DI APPLICAZIONE: Epanet in uDig

Sviluppo di un plugin di interfaccia al programma di calcolo acquedottistico EPANET.

Il motore e' stato sviluppato nei jgrasstools.

Una interfaccia grafica dedicata e' stata creata per uDig, per un utilizzo di tipo GIS.

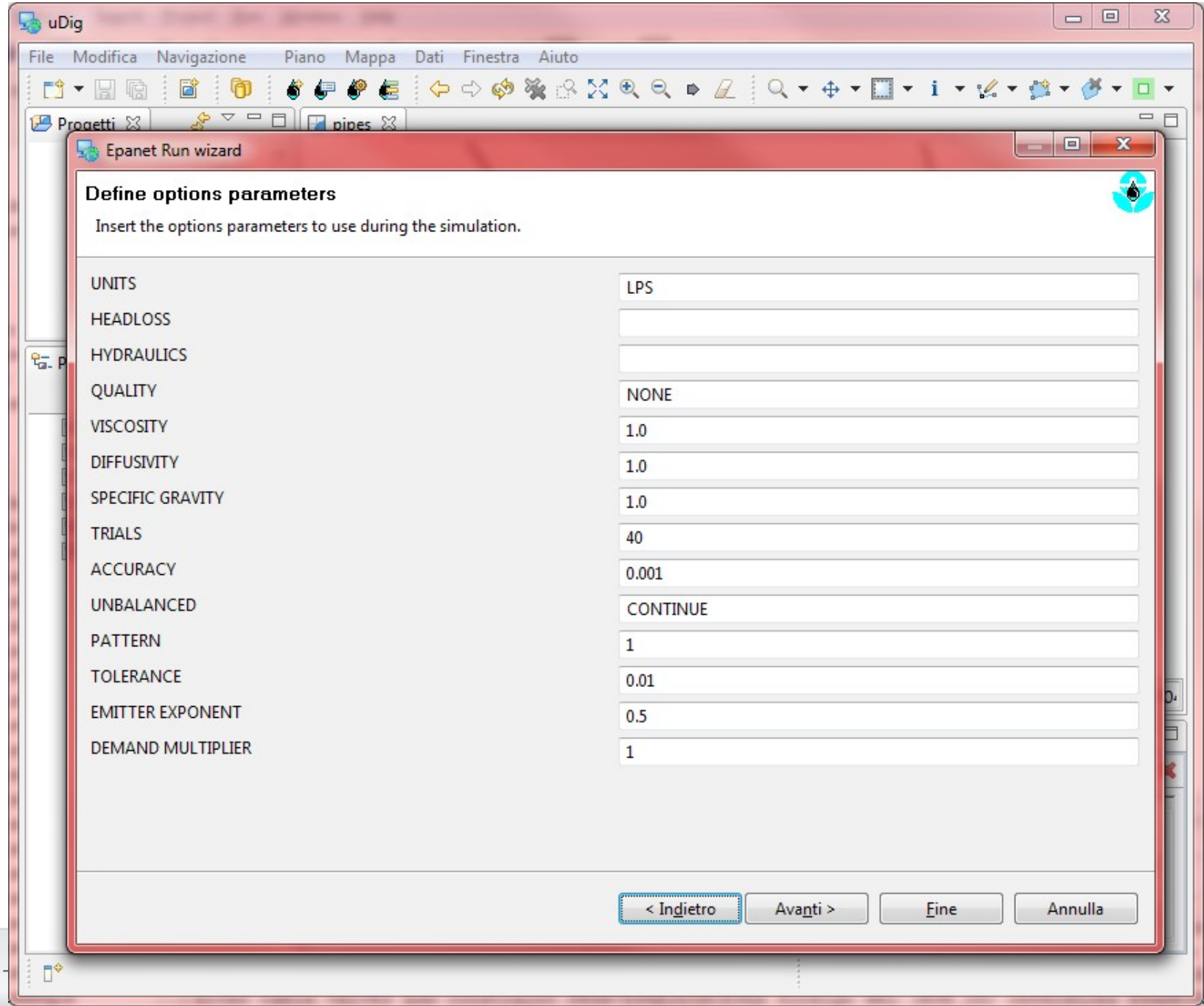
ESEMPI DI APPLICAZIONE: Epanet in uDig



- Generazione template
- Compilazione automatica di attributi
- Procedura guidata di esecuzione
- Visualizzatore risultati

ESEMPI DI APPLICAZIONE: Epanet in uDig

Procedura guidata inserimento dei parametri di simulazione e di esecuzione di EPANET

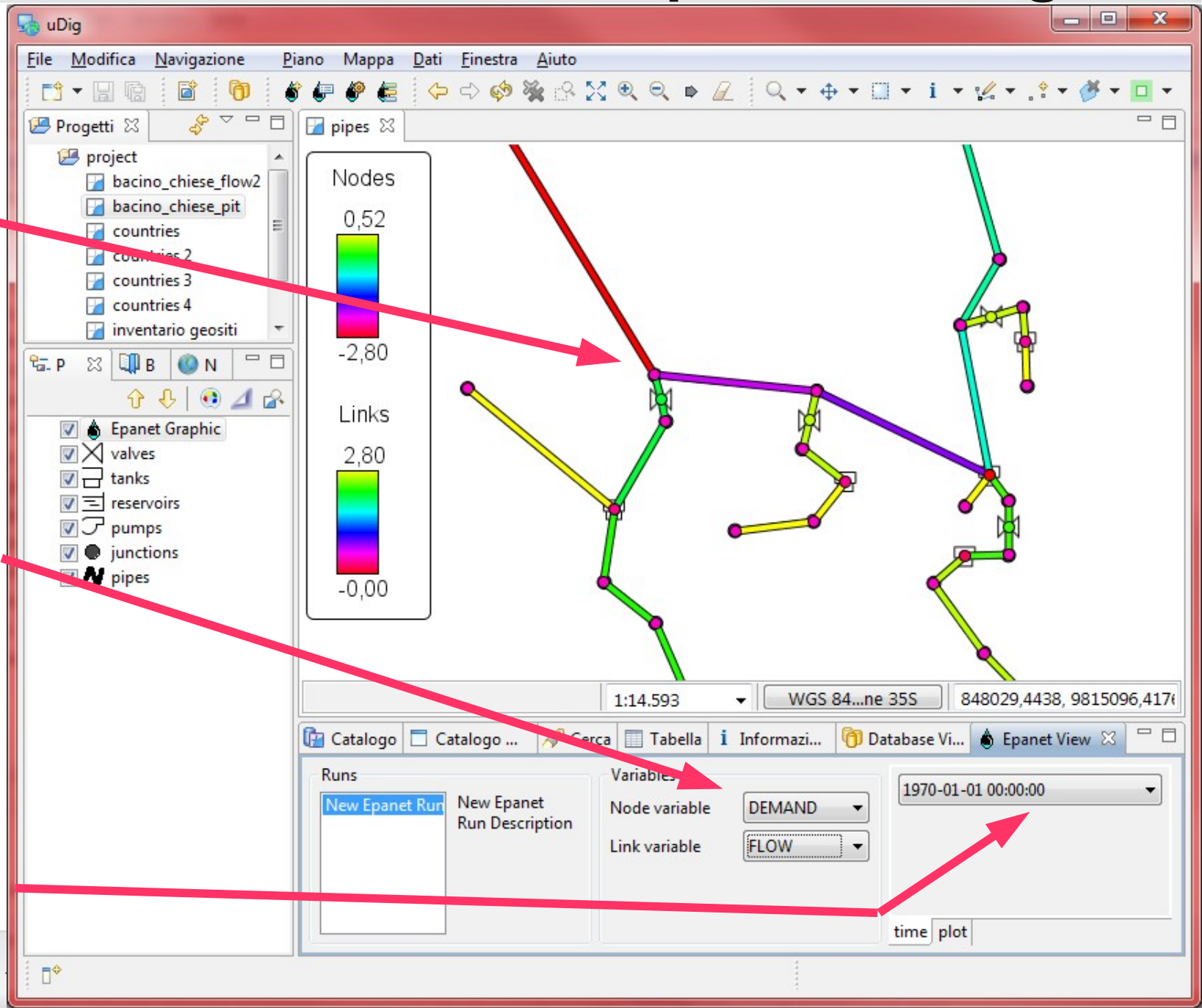


ESEMPI DI APPLICAZIONE: Epanet in uDig

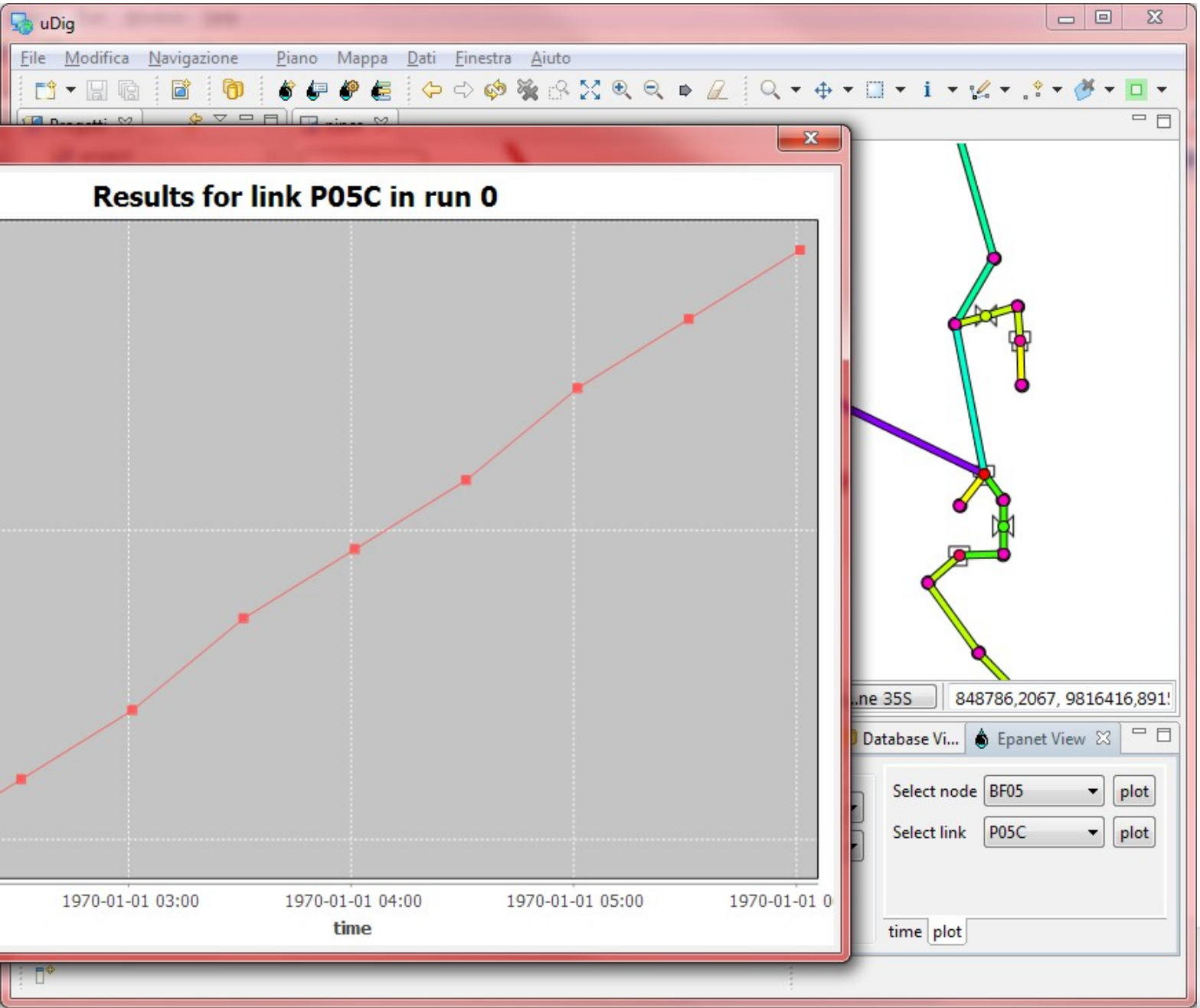
Visualizzazione dei risultati sovrapposti alla rete, nei nodi e nelle condotte

selezione della variabile da visualizzare

selezione dell'istante da visualizzare



ESEMPI DI APPLICAZIONE: Epanet in uDig



Visualizzazione dei risultati di un nodo o condotta per tutti gli istanti temporali

JGrasstools

http://www.jgrasstools.org
http://udig.refractions.net

SOME RIGHTS RESERVED

19 Novembre 2010