

A Java alternative to open source visualization - VisNow



Bartosz A. Borucki, Krzysztof S. Nowiński
Visual Analysis Laboratory
Interdisciplinary Centre for Mathematical and Computational Modelling
University of Warsaw, Poland



<http://visnow.icm.edu.pl>

<http://gitlab.com/ICM-VisLab/VisNow>

What is VisNow?

- Generic visualization and data analysis software
- Modular dataflow driven paradigm
- Focused on scientific visualization
- Implemented in Java
- Open source, GPL classpath exception license (LGPL equivalent)
- Support for large datasets
- Extensible
- Alternative proposition to Paraview or VisIt

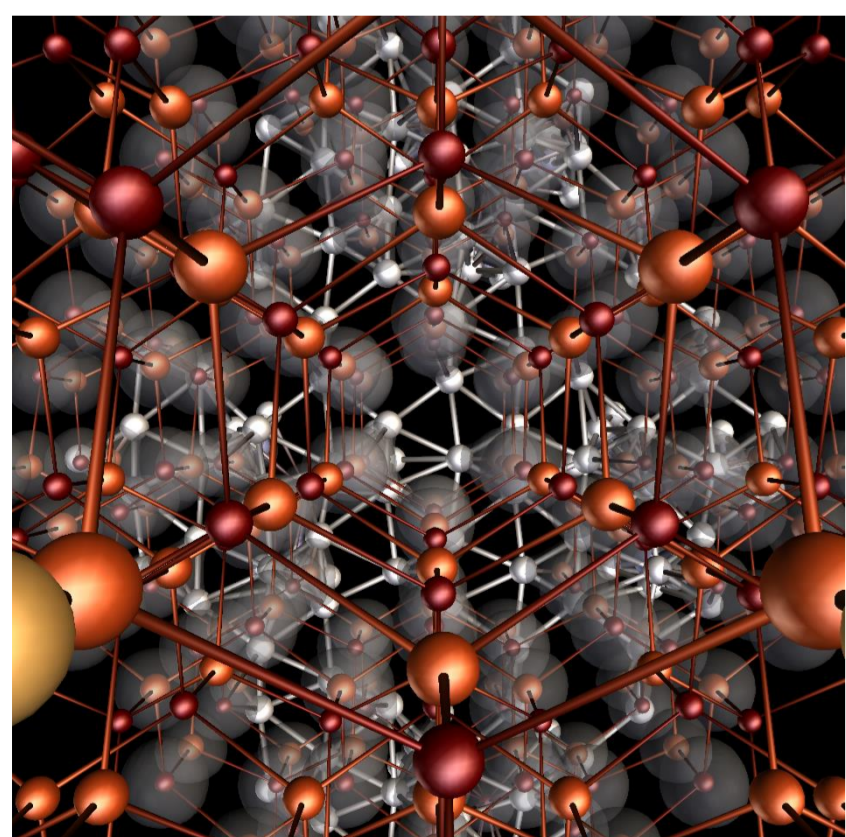
In dataflow paradigm modules serve as functional nodes providing certain functionality in the visualization pipeline (e.g. data access, filtration, mapping, presentation), and connections between modules represent data. The module-centric, dataflow driven concept of VisNow system is based on longtime experience of usage and development of AVS systems (AVS 3-5 and AVS Express), IBM Data Explorer and several other generic visualization systems, in contrast to the data centric paradigm of many popular systems (e.g. ParaView, VisIt). The module-centric approach can be more flexible in the case of complex dataflow with modules requiring multiple inputs.

VisNow application allows to create networks of modules for reading data, numerically processing them, converting to geometric objects (mapping) and presenting in viewer windows. Each module provides immediate graphical output that is automatically connected to the viewer module input, thus an immediate view of module processing results is available.

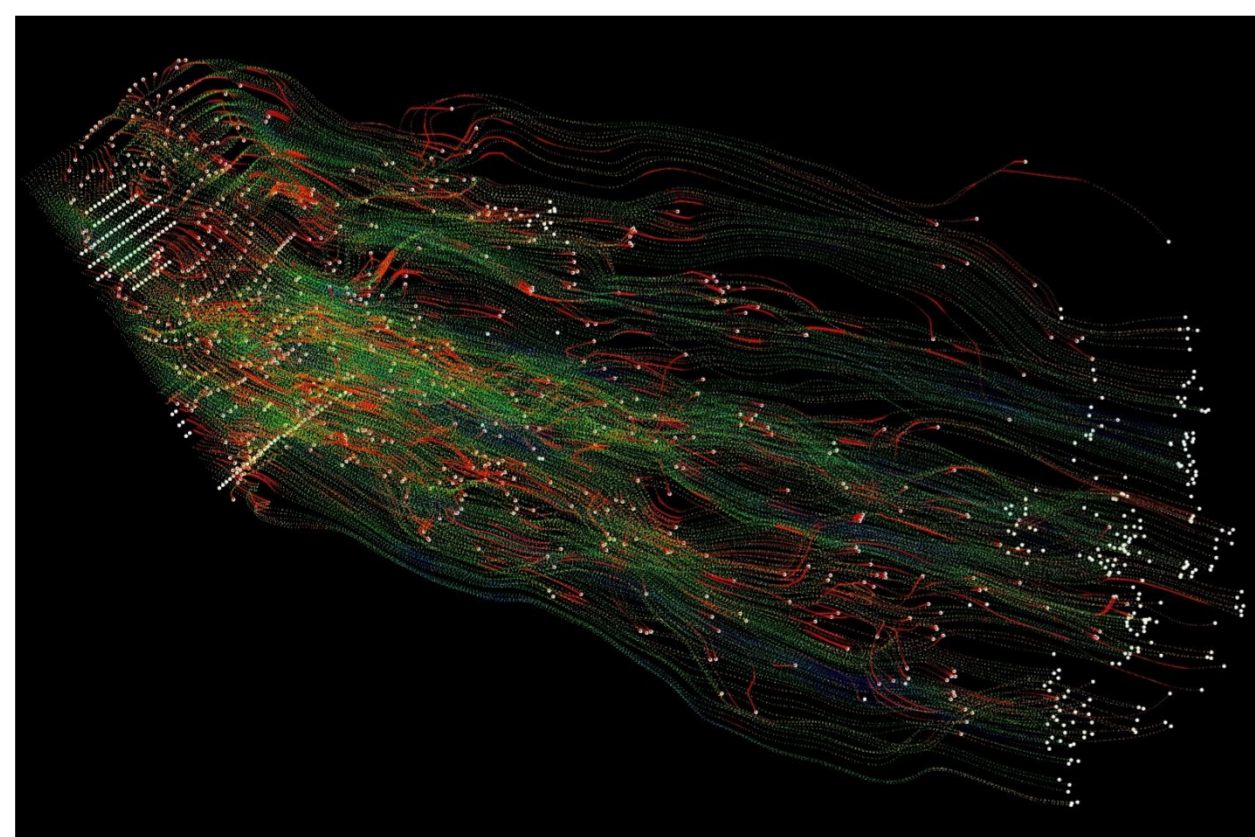
Motivation and design principles

- **High-level modules**
 - Complex modules with broad functionality – e.g. flow visualization
 - Contrary to majority of visualization systems with fine-grained technical modules
 - Ease of network creation and more legible network
- **Read-and-Watch**
 - Instant automatic visualization of data fields on each processing level
 - Visual representation incorporated in modules
 - Each mapping controlled independently
- **Reasonable default parameters**
 - Automated intelligent preparation of default parameters
 - Supports instant visualization

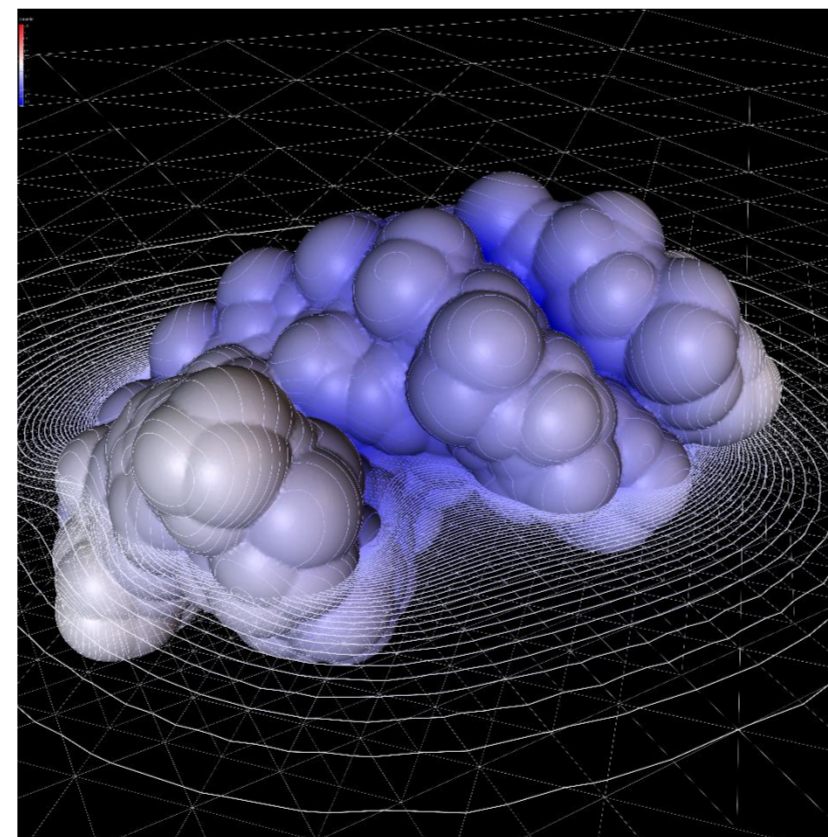
Applications and example visualizations



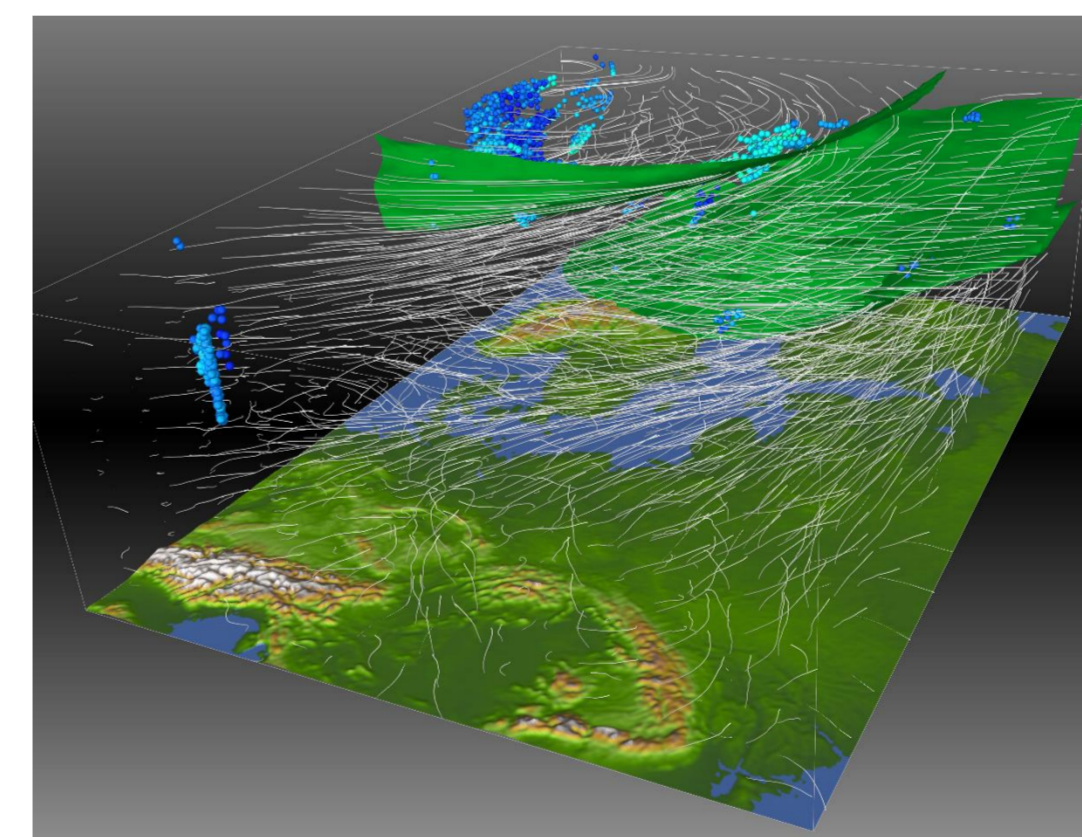
Solid physics – semiconductor structure and potential field.



Flow simulation through porous media (sand).



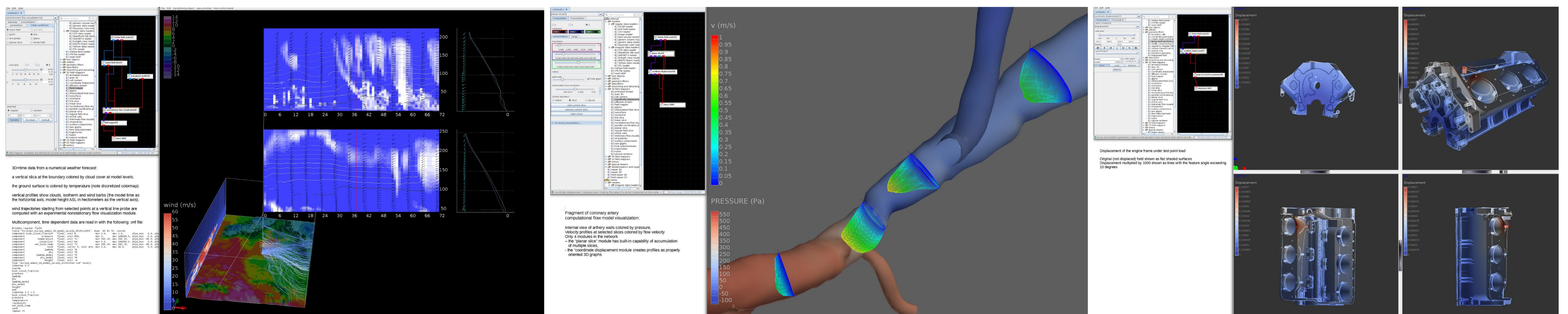
Chemistry – protein potential field.



Numerical weather forecast – wind, pressure and rain in time-space over Europe.



Medicine – blood flow analysis in coronary artery.



Functionality

- Generic representation of data („field”) based on **JSciC library**:
 - Geometry + structure + multivariate scalar or vector data components
 - Time dependency
 - Regular and Irregular (unstructured) field sub-types
- **Data access**:
 - VNF – VisNow field (own format based on JSciC library),
 - Basic VTK support, AVS fields, UCD, images, DICOM, OBJ, STL, EnSight, CSV
 - Generic metadata format (VNF header) to binary or ASCII data files
- **Processing**:
 - Remeshing, differential operations, transforms (e.g. FFT, Radon), interpolation, denoising
 - Data arithmetic with physical units support
 - Segmentation, skeletonization
- **Mapping**:
 - Colormapping, transparency mapping, texturing
 - Slicing, volume rendering, isosurfaces, isolines, glyph representations, streamlines, 2D/3D graphs
 - Correlation analysis, parallel coordinates
- **Presentation**:
 - 2D/3D viewer, 3D regular field viewer (orthoslices + 3D), engineering viewer
- **Output**:
 - Various data formats for field writing
 - Images and animations from viewer modules

VisNow functionality can be easily extended by programming new modules and adding module libraries (plugins), which can be created as separate Java projects (JAR files), also with different licensing. New libraries can be dynamically added and modified at runtime. Dataflow networks built with the help of interactive network editor can be wrapped into stand-alone dedicated application for the end users.

Conclusions

- Multiple similar products were and still are available in the market.
- VisNow was designed to somehow fill the gaps between different platforms, solve the major drawbacks and, above all, more directly respond to user expectations by providing lower entry level and more intuitive workflows.
- VisNow is based on the dataflow paradigm being at the same time more intuitive to inexperienced users and allowing experienced ones the creation of very complex visualization networks.
- Designed and created mainly for multicore desktops and workstations (with embedded parallelization and CUDA acceleration for most time consuming tasks), therefore, it does not support distributed and remote systems.
- Java was chosen not only for more productive development, but also to provide the same functionality on different operating systems, and to allow Java developers for simple integration with visualization libraries other than VTK.
- Multiple studies show that from both beginner and advanced user perspective, many tasks can be solved more easily in VisNow than in other platforms and both the user interface and workflows are more intuitive.
- Future development is planned in the direction of distributed client-server architecture, mobile and web interfaces and even more simplified user interface.