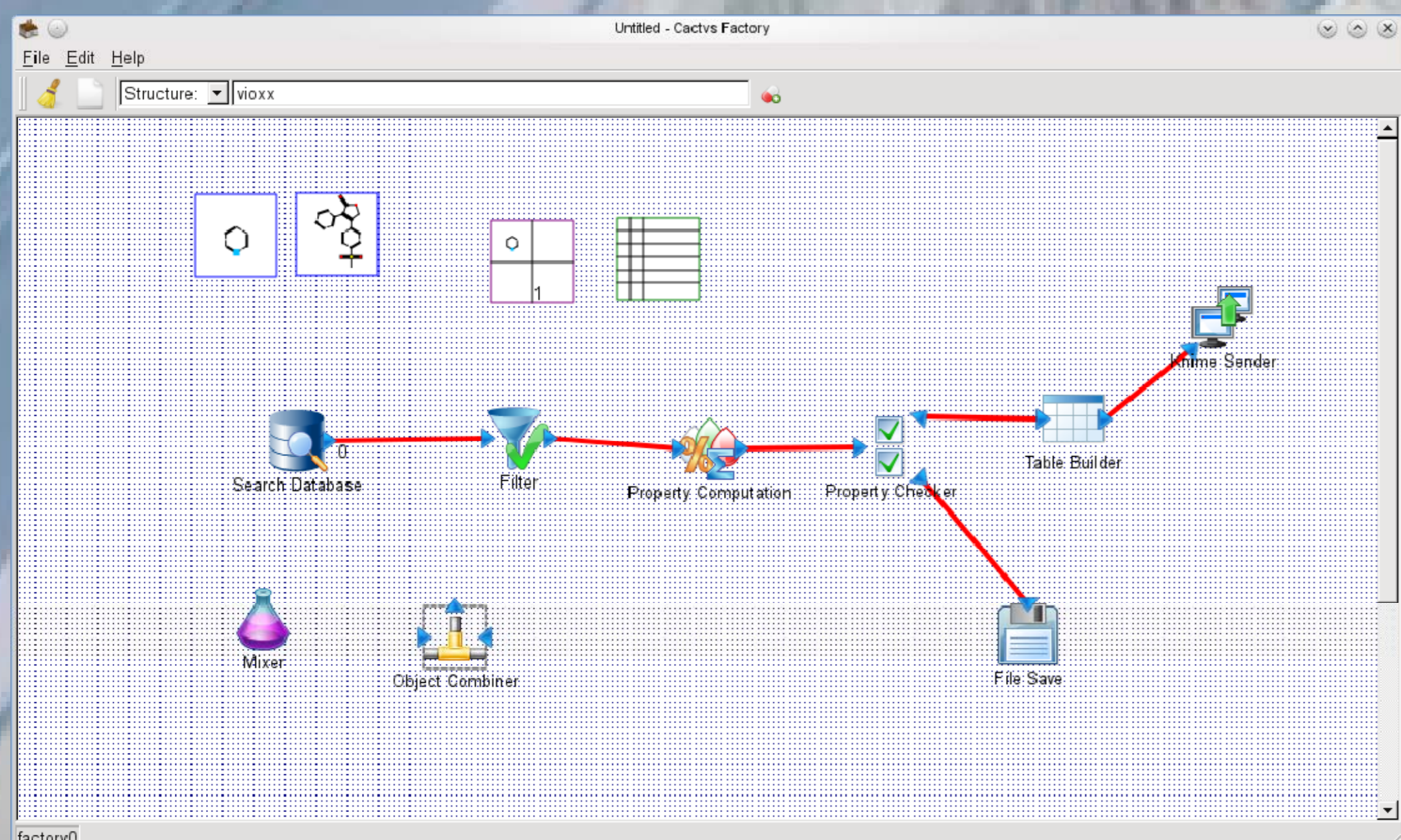


Revisiting the Dataflow Paradigm for Chemical Information Processing



Dataflow processing systems, such as Pipeline Pilot or KNIME, are now a major trend in chemical information processing. Established solutions feature comprehensive node collections. Many routine tasks are effectively addressed by combining standard building blocks and configuring their parameters. On the other hand, while these systems sometimes have specialist-accessible development and scripting capabilities, writing custom nodes beyond minor variants of existing tools is typically difficult and possible operations are constrained by the framework.

Non-graphical chemistry scripting environments, such as Cactvs/Tcl, OpenEye/Python or the Indigo multi-language suite are generally more powerful in this respect, but have steep learning curves and issues with code reusability.

Revisiting our pioneering 1995 work on adapting dataflow principles for chemical information processing, and building on our experience with the Cactvs chemistry data processing scripting toolkit, we have now implemented a novel software design which combines the benefits of free scripting with the convenience of dataflow solutions.

Pipeline software streams are similar to tubing in a refinery in that nobody wants to open them up and touch the stuff inside. Pipeline software processes continuous data streams, with minimal user interaction once a configuration is set up.

Our new software **Cactvs/Factory** in contrast operates like an automated factory floor, where identifiable, isolated objects move via conveyor belts from one processing station to the next – and may be taken out, stored, examined and resubmitted at any intermediate step.

The four standard processed chemistry objects are structures, reactions, datasets, and tables, each with arbitrary attached properties.

Cactvs/Factory is internally implemented as a shell around the generic Cactvs scripting toolkit.

All items in this environment are standard scriptable Cactvs objects. This includes the newly introduced object types of processing stations, conveyors and factory floors.

These, and all standard chemistry objects may be accessed, created, destroyed and manipulated by script commands. The graphical user interface is optional and for convenience: It mostly issues commands which could also be typed on the console.

Processing stations are at their core Cactvs script snippets. Every node executes at least one independent, sandboxed script interpreter in its own thread. If the order of processed data does not matter, additional threads with more interpreters can be started on more expensive nodes.

Every station interpreter has access to all Cactvs toolkit extensions, such as property definitions and their computational modules, I/O modules for chemistry files and table data, command extensions, etc.

A smart configuration panel helps in the definition of custom stations. Various debugging facilities, down to a text console with direct script access to any object on a factory floor, facilitate rapid development of custom solutions.

Nodes or factory set-ups are stored as files in XML format, are platform-independent and transportable.

Data objects may be inspected anytime as they are being transferred from station to station, or onto the factory floor. Viewers and editors for structures, reactions, datasets and tables are provided.

Defective objects, which for example trigger an error in a processing station, do not bring the complete set-up to a standstill, but are rather automatically ejected from the station onto the factory floor, where they may be inspected, debugged and fixed, or discarded.

Stations may be connected for automatic transfer of generated or processed objects via a conveyor belt system running in an independent thread.

Station ports are configured to accept certain types of objects on input, and deposit their output objects into output port buffers, where they may be inspected, collected for further processing, or undergo standard predefined operations such as destruction, sampling or dataset assembly.

Because it is an object flow, belt connections cannot fork, but designs such as mixed object types in a flow, directly reconnecting the output of a station to one of its inputs or setting up larger loops and networks are possible.

The software is primarily designed to process chemical entities. For statistical number crunching, data is usually collected in a table object that is exported to external analysis systems.

Cactvs/Factory factory floors can be connected for bidirectional, interactive data exchange to KNIME workspaces by means of a collection of Xemistry KNIME nodes.

Doing chemistry in Cactvs/Factory is generally much faster than in KNIME, by a factor of 100 or more for certain tasks.

The toolkit can read and write the native binary KNIME data table format, including its structure, reaction and image cell types. This enables convenient further processing without the need for complex set-up of input data in KNIME.

Acknowledgement: A part of the initial development of this software was supported by the Nicklaus group at NIH Frederick, MD, USA.