chem-bla-ics

Scripting JChemPaint

Egon Willighagen

Published November 20, 2008

Citation

Willighagen, E. (n.d.). In *chem-bla-ics*. chem-bla-ics. https://doi.org/10.59350/yhdxa-ft783

Today and tomorrow, Stefan, Gilleain, Arvid and I are having a JChemPaint Developers Workshop in Uppsala, to sprint the development of JChemPaint3, for which Niels layed out the foundation already a long time ago.

Gilleain and Arvid are merging their branches into a single code base, while Stefan is working on the Swing application and applet. The Bioclipse SWT-based widget is being developed for Bioclipse2.

The new design separates widget/graphics toolkit specifics from the chemical drawing and editing logic. Regarding the editing functionality, this basically comes down to have a semantically meaningful edit API. This allows us to convert both Swing and SWT mouse events into things like `addAtom("C", atom)`, which would add a carbon to an already existing *atom*. However, without too much phantasy, it allows adding a scripting language. This is what I have been working on. Right now, the following API is available from the Bioclipse2 JavaScript console (via the *jcp* namespace, in random order):

- ICDKMolecule jcp.getModel()
- IAtom getClosestAtom(Point2d)
- setModel(ICDKMolecule) *(for really fancy things)*
- removeAtom(IAtom)
- IBond getClosestBond(Point2d)
- updateView() *(all edit command issue this automatically)*
- addAtom(String,Point2d)
- addAtom(String,IAtom) *(which works out coordinates automatically)*
- Point2d newPoint2d(double,double)
- updateImplicitHydrogenCounts()
- moveTo(IAtom, Point2d)
- setSymbol(IAtom,String)
- setCharge(IAtom,int)
- setMassNumber(IAtom,int)
- addBond(IAtom,IAtom)
- moveTo(IBond,Point2d)
- setOrder(IBond,IBond.Order)
- setWedgeType(IBond,int)
- IBond.Order getOrder(int)
- zap() *(sort of `sudo rm -Rf /`*)
- cleanup() *(calculate 2D coordinates from scratch)*
- addRing(IAtom,int)
- addPhenyl(IAtom)

This API (many more method will follow) is not really aimed at the end user, who will simply point and click. The goal of this scripting language is, at least at this moment, to test the underlying implementation using Bioclipse. Future applications, however, may include simple scripts which use some logic to convert the editor content. For example, replacing a t-butyl

fragment into a pseudo atom "t-Bu". The key thing to remember, is that this will allow Bioclipse to have non-CDK-based programs act on the JChemPaint editor content (e.g. using `getModel()` and `setModel(ICDKMolecule)`). More on that later.

A simple script could look like: Or, as screenshot: