

JChemPaint - Using the Collaborative Forces of the Internet to Develop a Free Editor for 2D Chemical Structures

Stefan Krause¹, Egon Willighagen² and Christoph Steinbeck^{1*}

¹Max Planck Institute of Chemical Ecology, Tatzendpromenade 1a, 07745 Jena, Germany
Tel.: +49(0)3641 643644, Fax: +49(0)3641 643665, E-mail: steinbeck@ice.mpg.de

²Department of Chemistry, University of Nijmegen, The Netherlands

* Author to whom correspondence should be addressed.

Presented at the Third International Electronic Conference on Synthetic Organic Chemistry (ECSOC-3, <http://reprints.net/ecsoc-3/E0004/e0004.html>), September 1-30, 1999.

Received: 26 January 2000 / Accepted: 27 January 2000 / Published: 28 January 2000

Recommended for publication by Shu-Kun Lin (lin@mdpi.org)
and Peter Ertl (peter.ertl@pharma.Novartis.com)

Abstract: The open source program for drawing 2D chemical structures JChemPaint, its current features, its envisioned further development and the principles enabling researchers and students at places all over the world to collaboratively develop such a program are described.

Keywords: Chemical Markup Language, Java, Open Source, Structure Editor, Structure Diagram Generation.

Introduction

2D chemical structure editors are central tools in fields like chemoinformatics, computational chemistry and synthetic chemistry. No matter if one wants to submit a structure query to a database,

prepare a starting structure for molecular modeling, draw a set of structures for good lists and bad lists in Computer Assisted Structure Elucidation (CASE) or just sketch a structure or reaction scheme for a publication - in all of these cases the starting point is opening a structure editor. Programs for drawing chemical structures are abundant and a number of formerly commercial programs in this area are now available free of charge for non-profit use, like Isis/Draw (MDL Information Systems, Inc., <http://www.mdli.com>). Nevertheless, there are no state-of-the-art programs available in source code with a free licensing scheme, thus allowing researchers to adapt and embed them into their own programs without paying license fees. Such an open source structure editor would be of interest for many reasons, of which just not having to pay for it is certainly the weakest argument. Firstly, it would ease the work of all those developers who need to be able to change and adapt the source code of a module they use in order to integrate it into their projects. The authors of programs that calculate NMR shifts or generate the IUPAC name for a given structure would not have to rewrite this standard piece of software again and again. Secondly, open source projects are known to solve problems with faulty software in a short time since bugs are much more easily found and improvements are much more easily made if everyone can have a look at the source code. The development of such a project would ideally be done in Java with its unique features of being platform independent, easy to learn, highly structured and well integrated with web technology, enabling the use of JChemPaint for all kinds of web based projects, like electronic publishing. Thus, the intriguing characteristics of the open source paradigm, the introduction of Java with its platform independence as well as the surprising lack of a free, open source, platform independent structure editor made it desirable to initiate the JChemPaint project.

Results and Discussion

Two of us (CS, SK) started the JChemPaint project and began writing an open source program for drawing 2D chemical structures. It was decided that Java 2, also known as Java 1.2, should be used for the development. We soon discovered that a complementary program for displaying 3D structures, JMol [1], had been developed within the Open Science Project of Dan Gezelter at Columbia University [2]. It quickly became our vision that both programs could form a comprehensive system for handling 2D and 3D chemical structures like it is found in, for example, the commercial ChemOffice suite (CambridgeSoft Corporation, <http://www.camsoft.com>). EW then joined both teams and added support for structure I/O in Chemical Markup Language (CML) [3].

JChemPaint already offers a great amount of functionality. Figure 1 shows a steroid drawn with the use of ring templates. Currently supported features are:

1. A subset of the regular drawing features of commercial programs, such as
 - drawing of single, double and triple bonds
 - stereo descriptors
 - deletion of bonds and atoms

- templates for rings of size 3 to 8
 - one click attachment of templates to an atom or a bond
 - flipping and rotating selected parts of the molecule
2. Loading and saving structures in Chemical Markup Language (CML) and as MDL Molfiles [4]
 3. Automated Structure Layout, also known as Structure Diagram Generation.
 4. Loading structures from the “Dictionary of Organic Chemistry” [5] using the CAS Registry Number.

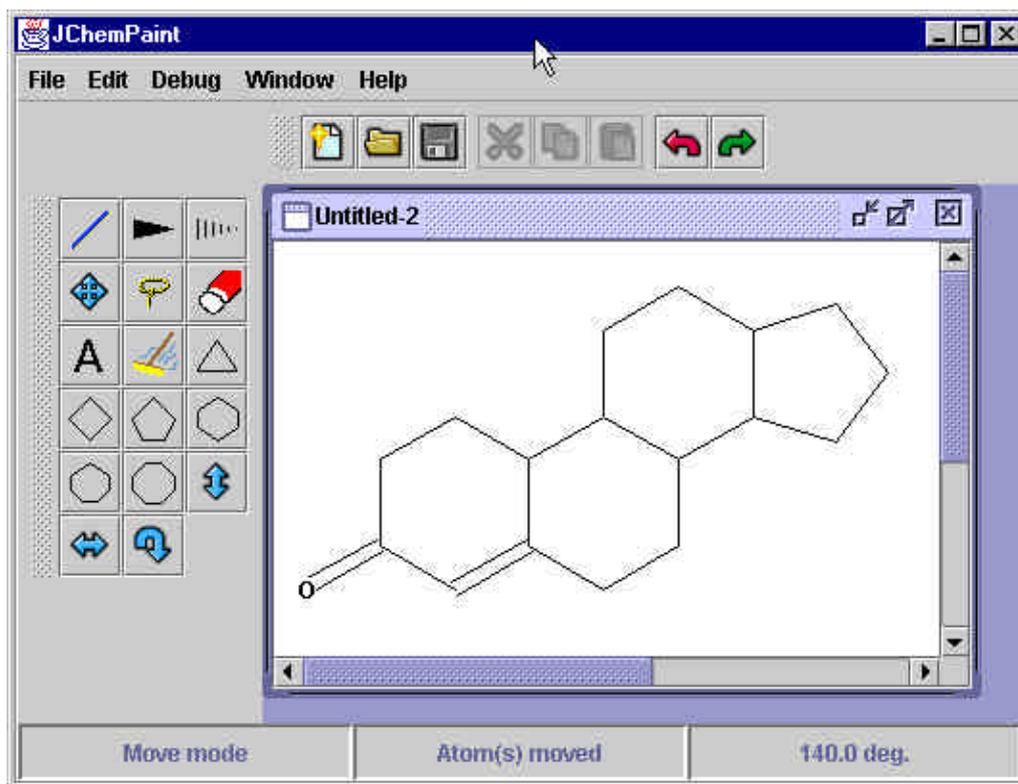


Figure 1. A screenshot of JChemPaint at work.

Taking into account the fairly large number of programs available in this field, there would seem to be no great challenges in designing such a system. However, some aspects are tricky to program and pose interesting problems, for example, for student education. JChemPaint, written in an object oriented programming language, with its clear and modular design, and its source code available to everyone, seems to be the ideal playground for trying new ideas and optimizing existing solutions. An example case in which JChemPaint can be used for educational purposes is its capability of handling Chemical Markup Language (CML), the upcoming universal language for managing chemical information [3,5]. CML is an extension of the Extensible Markup Language (XML) [6] and is likely to have a large impact on the way of how chemists encode their chemical information. The process of designing CML is not yet closed and it is thus especially exiting to have look at, or even take part in, the on-

going development. Details on the CML implementation in JChemPaint and JMol can be found in [7].

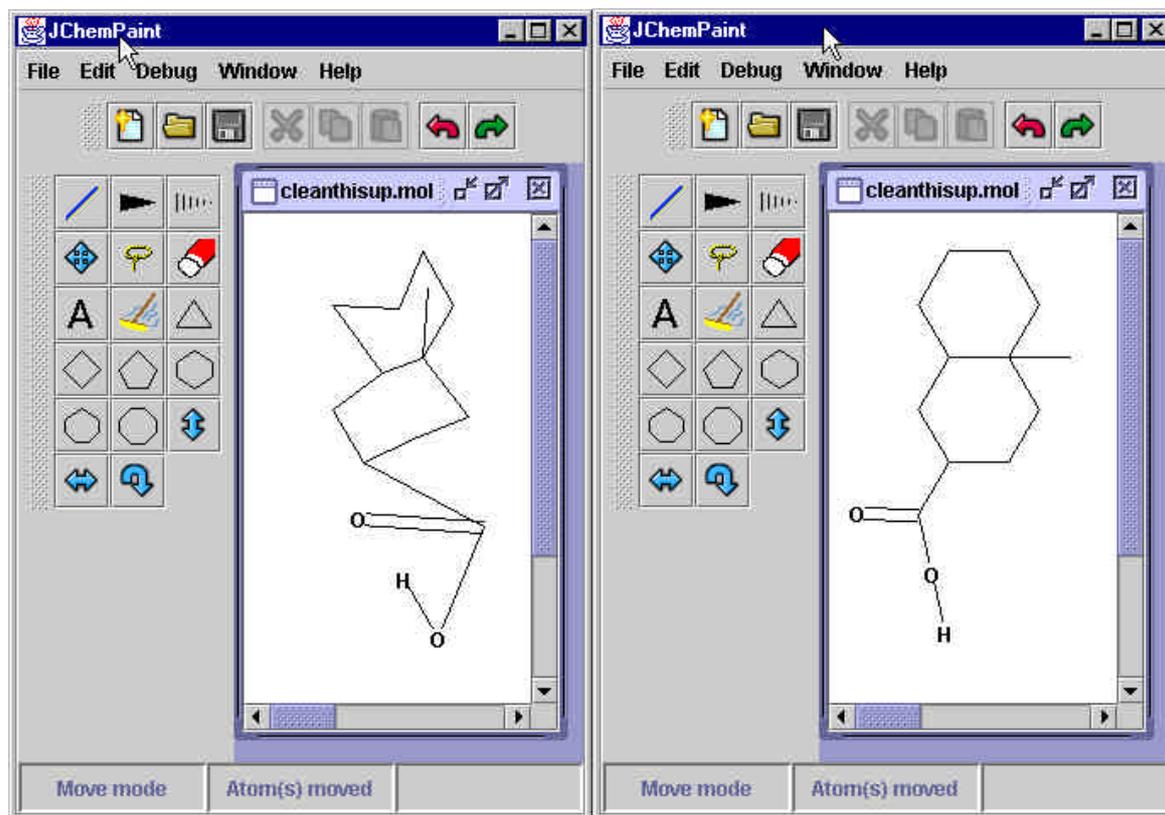


Figure 2. Before and after - the effect of a JMDraw clean up.

A prominent case for which an implementation exists in JChemPaint is the solution to the Structure Diagram Generation problem - comprehensively summarized in the review of Harold E. Helson at CambridgeSoft [8]. Here, a molecular graph, with or without layout information, for which a cleanup is desirable, is subjected to an algorithm which lay outs each atom in such way that the resulting picture of the molecule complies with the conventions used by chemists to hand draw that structure. In JChemPaint we use a the Java module JMDraw, written by one of us (CS), which is based on the C program MDraw by Ugi and coworkers [9]. While the resulting layout is sufficient in many cases, there is still plenty of room for improvements and JChemPaint's open source is the ideal basis for that.

The Development Model

The development of JChemPaint is maintained via the Concurrent Versions System (CVS) model [10], a widely used system with a client-server architecture that allows users to independently and concurrently work on even the same parts of the source code by checking out personal copies of the software from the central repository, making their changes and checking in again the modified source code. The CVS system then tries to merge the independently modified versions of the source into the

repository and only in rare cases requires intervention by the user for this purpose. Developers can synchronize their local versions with the central repository automatically over the Internet, making distributed development possible. Textual communication between the developers is organized via the JChemPaint web pages and an electronic mailing list. New versions of the program are released to the public frequently and early, as recommended by Eric Raymond in his brilliant analyses of the principles driving the open source development concept [11]. Each announcement causes a number of interested and potential new co-developers to join the developers' mailing lists and a number of them contribute by discussing questions of program design.

As pointed out in the introduction of this article, an important aspect of the open source paradigm was that the source code can be adapted and embedded in other programs. This is also the case for JChemPaint. An ongoing Open Science project is the CML Server project [12] for which web server software will be developed to publish chemical data contained in, for example, chemical articles or chemical databases. Software written for the JChemPaint project will certainly be embedded in this project and maybe even adapted as new requirements become clear. New software developed can, in turn, be embedded in JChemPaint again, thus making full use of the open source ideals.

Conclusion and Outlook

We have described the program JChemPaint, a 2D molecular structure editor. While the program itself as well as most of the underlying algorithms are not scientifically thrilling material, it is its development model and the wide usability of the software that might attract the attention of a potentially large group of users and of some highly welcome new co-developers. We have also shown the powerful distributed development model by which JChemPaint is developed, how open sourcing can help the development of this project and how it can help the development of other software.

A great number of improvements and new features still await implementation. Professional quality output is, for example, not yet possible nor is adaptation to different types of layout, to mention only two possible fields of improvement. An obvious new feature is the implementation of an open source algorithm for conversion to and from structural names conforming to the IUPAC nomenclature rules. A lot of work also remains to be done on the interface with the JMol software. For example, a 3D model builder, the 3D analogue of JMDraw, is the first thing to be mentioned. This, needless to say, is a whole new open source project by itself. The experiences from other open source projects show that a critical mass of working features must be implemented in order to attract new contributors. We hope that, as the program grows, the developer community will also.

References and Notes

1. JMol can be downloaded from <http://www.openscience.org/jmol/>.
2. The Open Science project website is <http://www.openscience.org/>.

3. Murray-Rust, P.; Rzepa, H. S. Chemical Markup, XML, and the Worldwide Web. 1. Basic Principles. *Journal of Chemical Information and Computer Sciences* **1999**, 928-942. For constantly updated documentation of Chemical Markup Language (CML) please see <http://www.xml-cml.org/>.
4. Dalby, A.; Nourse, J. G.; Hounshell, W.D.; Gushurst, A.K.; Grier, D.L.; Leland, B.A.; Laufer, J.; Description of Several Chemical Structure File Formats Used by Computer Programs Developed at Molecular Design Limited. *Journal of Chemical Information and Computer Sciences* **1992**, 244. An updated online version of this document can be found on <http://www.mdli.com/downloads/literature/ctfile.pdf>
5. The “Dictionary of Organic Chemistry” is found on <http://www.sci.kun.nl/woc/>
6. XML is a recommendation written by the World Wide Web Consortium (W3C). Info on this standard can be found at <http://www.w3.org/TR/1998/REC-xml-19980210>.
7. Willighagen, E. L. “Processing CML conventions in Java”, http://www.openscience.org/~egonw/-cml/cml_conventions.html
8. Helson, H. E. in *Reviews in Computational Chemistry*; Lipkowitz, K. B.; Boyd, D. B. Eds.; Wiley-VCH: New York, 1999; Vol. 13, pp. 313-398. Structure Diagram Generation.
9. Bley, K.; Brandt, J.; Dengler, A.; Frank R.; Ugi, I. Constitutional Formulae generated from Connectivity Information: the Program MDRAW. *Journal of Chemical Research (M)* **1991**, 2601-2689.
10. Please see <http://www.sourceforge.com/CVS/> for information on the CVS system.
11. E. S. Raymond’s “The Cathedral and the Bazaar”, “Homesteading the Noosphere” and other related articles are found on <http://www.tuxedo.org/>.
12. The CML Server project is currently in a preliminary state. Information on this project is located at <http://www.openscience.org/~egonw/cmlserver/>.