The Third International Symposium on Optimization and Systems Biology (OSB'09)
Zhangjiajie, China, September 20–22, 2009
Copyright © 2009 ORSC & APORC, pp. 402–409

# ChemIME: An Input Method Engine for Chemical Compounds[*]

Haruka Takeuchi Xu Yang Shin-ya Takane[†]

Department of Information Systems Engineering, Osaka Sangyo University,
Osaka 574-8530, Japan

**Abstract** In the field of materials sciences, it is often needed to convert a compound name into corresponding molecular equations, 2D or 3D molecular structures. We present a concept and a prototype implementation of the input method engine that can convert various compound names (given as an IUPAC name) into corresponding molecular equations by using Java and the Java Input Method Framework. We also extend to give various 2D or 3D coordinate formats (MDL mol, CML, and Gaussian input file) with the help of Chemistry Development Kit (CDK) and some external programs.

**Keywords** Java; software; Input Method Framework (IMF); Chemistry Development Kit (CDK)

## 1 Introduction

In the field of materials sciences, especially that of chemistry, which mainly treats various compounds, it is often needed to convert a compound name into corresponding molecular equations, 2D or 3D molecular structures. For such purpose, there exist some useful commercially available tools to generate the molecular structure or rational formula from an IUPAC name [1, 2], for example, ChemDraw [3] and Chemistry 4D-Draw [4]. These software packages, however, are separate applications for each platform and cannot be invoked on the spot from other applications. To our knowledge, there is no application that has the function of in-line conversion for chemical compound names. Our idea is that using an input method mechanism that allows to input characters other than Roman alphabets with a standard keyboard, it is possible to implement such a tool for material scientists more effectively. However, if the system uses only a usual key-value (one-to-one) dictionary that stores a compound name and corresponding formulas, the dictionary becomes huge and cannot manage it in practice.

In this paper we present a concept and a prototype implementation of the input method engine that can analyze the functional groups and convert various compound names (given as an IUPAC name) into corresponding molecular equations by using Java and the Java Input Method Framework. We also extend to give various 2D or 3D coordinate formats (MDL mol, CML, and Gaussian input file) with the help of open source Chemistry Development Kit (CDK) and iText libraries, and some external programs.

[*] This work was supported in part by JSPS under JSPS-NSFC collaboration project.

[†] Corresponding author. Email: takane@ise.osaka-sandai.ac.jp.

# 2 Methods

## 2.1 Java

Java is an object-oriented programming language developed by Sun Microsystems, Inc. [5] and was designed by reference to other languages, especially C++. It is often called a platform independent or multi-platform language with the catchphrase "Write once, run anywhere".

Java is not a language that generates an executable code directly by compilation process, but generates an intermediate code called "bytecode" firstly, and then it can be executed by the bytecode interpreter called Java Virtual Machine. Since the Java VMs are available for most operating system (OS), including Mac OS, Linux, and Windows, the same compiled bytecode can run on most platforms without any modification. Table 1 shows software development environment used in this study.

Table 1: Software environment used in this study

| | |
|---|---|
| OS | Windows XP SP2 / Mac OS X 10.4 and 10.5 |
| Java | JDK 1.5.0 / JDK 1.6.0 (Java SE 6.0) |
| CDK | CDK 20060714 |
| iText | ver. 1.4.7 |
| JChemPaint | ver. 1.1 |
| Jmol | ver. 10.3 |

## 2.2 Input Method

The input method engine (or editor) is software that is used for input of characters of languages like Japanese, Chinese, or Korean, which cannot input the characters with a standard Western keyboard. Of course it is infeasible to make a keyboard that allocate all characters necessary to these languages, because they need thousands of keys. In general each (non-Western) character is assigned to the combination of alphabetical keys (more than one), and after pressing a special key (usually assigned to the SPACE key), the entered characters are converted to objective character(s). If there exist multiple candidates, the user needs to choose an item from the list (conversion list). This process is called conversion (or transformation). After choosing the item, another special key (usually assigned to the ENTER key) is pressed to confirm the conversion.

## 2.3 Java Input Method Framework

Java Input Method Framework (IMF) can help the interaction between text components and the input method during text input and/or editing process. It was implemented since Java2 SDK (JDK 1.2). Since the previous versions could not support a standard method to treat data from the input method, developers just had to implement machine-dependent codes. IMF made the development of the input method that supports all Java applications (using the Swing text component) possible. There are two layers in IMF. One is the input method client Application Programming Interface (API) and the other is the input method engine Service Provider Interface (SPI). The input method client API enables client components to implement text input user interfaces such as on-the-spot input, whereas the engine SPI enables the implementation of the input method itself (Fig.1).

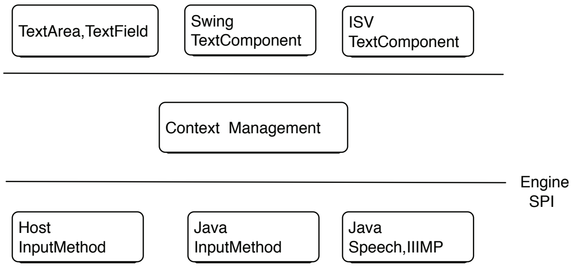


Figure 1: Structure of input method framework

## 2.4 Chemistry Development Kit (CDK)

CDK [6] is an open source Java library for handling chemical information. It can treat the information not as an original format, but also treat various formats such as Chemical Markup Language (CML) [7], MDL mol [8], PDB [9] and so on, and linear notations such as SMILES [10] and IUPAC International Chemical Identifier (InChI) [11]. Furthermore, it also supports the input file formats for computational chemistry software such as Gaussian [12]. The present ChemIME only supports MDL mol, CML, and Gaussian input formats. The binary / source codes of CDK are available from the web site [13]. The classes and packages used in this work are shown in Fig.2.

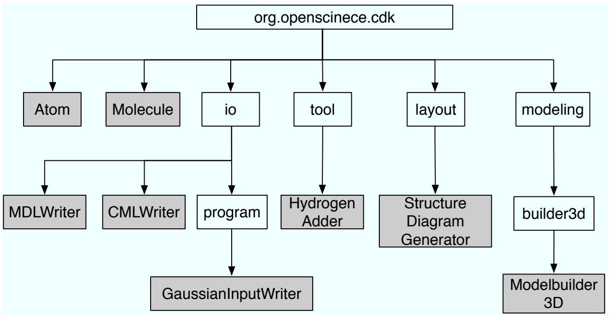


Figure 2: Classes and packages of CDK used in this work. The classes are shown in gray shaded boxes.

## 2.5 iText

Although iText is an open source Java library for generating Portable Document Format (PDF) files, it also contains the libraries for treating Hyper Text Markup Language

(HTML), Extensible Markup Language (XML), and Rich Text Format (RTF) files. We used it to save the contents of the text editor as RTF files in this study. The purpose of the RTF in this study is to display the subscript characters (like as Ąg4Ąh in $CH_4$) properly for chemical formulas on the text editor. This library is available from the web site [14].

### 2.6 IUPAC nomenclature and conversion algorithm

The IUPAC nomenclature is a systematic rule of naming chemical compounds agreed on by the International Union of Pure and Applied Chemistry (IUPAC). In this study, we only considered for organic compounds. In IUPAC nomenclature, there are six nomenclatures (substitutive nomenclature, radicofunctional nomenclature, additive nomenclature, subtractive nomenclature, conjunctive nomenclature, and replacement nomenclature) for organic compounds. The substitutive nomenclature is mainly subjected to this study. In the case of the substitutive nomenclature, the basic algorithm of the conversion is:

1. to parse input data
2. to determine the length of carbon main chain
3. to determine the position of double and / or triple bonds
4. to determine the position and types of the functional groups

At present, this system can recognize about 100 functional groups. Thus, by combining these groups, more than 10,000 compounds can be recognized on this system. The lists of available functional groups are shown in Table 2. The group name in the parentheses in Table 2 are recognized by the parser. If the name consists of two words (for example, sulfenic acid), the space should be replaced with an underscore character (e.g. sulfenic_acid) as an input string in the present implementation.

## 3 Results and Discussion

### 3.1 ChemText

The input method engine we developed should work on any Java application that supports the Swing text component. Therefore we also developed a text editor as a sample client Java application. As shown in Fig.4, the editor can support to change the fonts, styles, colours, and alignments. The tabbed document interface (TDI) is introduced to edit multiple documents with one window at the same time. Moreover, RTF is supported as output format by use of the iText library.

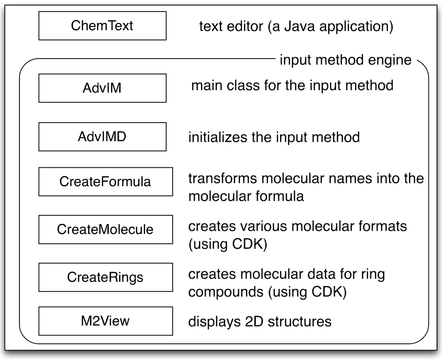
### 3.2 ChemIME

The Java classes we implemented in this work are illustrated in Fig.3. We have implemented a prototype of the input method that can convert basic IUPAC names to molecular formulas by entering the SPACE key at first and can further select other formulas, 2D view, and 3D view from a selection list. Fig.4 shows the snapshot of the input method invoked on ChemText.

Two features are implemented in combination with the text editor. One is to change the style of the number of molecular formulas into subscript automatically when the SPACE key is pressed to confirm, and the other is to paste the image of 2D structures in the document when selected the "2D structure view". These functions are only available when ChemText is used as a client editor.

Table 2: Lists of available functional groups. The group name in the parentheses are recognized by the parser.

| |
|---|
| alkane (-ane), alkene (-ene), alkyne (-yne), alcohol (-ol), hydroxy group (hydroxy-), calbonic acid (-oic acid), carboxy group (carboxy-), aldehyde(-al,-carbaldehyde), formyl (formyl-), ether (-ether), alkyl group (-yl), acyl group (-oyl), alkoxy group (-oxy), ketone (-one, -ketone) |
| halogen group (fluoro-, chloro-, bromo-, iodo-) |
| amine (-amine), amide (-amide), imine (-imine), nitro group (nitro-), nitroso group (nitroso-), amino group (amino-), diazo group (diazo-), azo group (-diazene), aminium group (-aminium), (diazonium), (-nitrile / cyano-), azido group(azido-) (isocyano-), (cyanate), (isocyanate) |
| thioamide (-thioamide), thiol (-thiol), mercapto group (mercapto-), thioaldehyde (-thial), thioketone (-thione), sulfonic acid (-sulfonic acid), sulfenic acid (-sulfenic acid), sulfinic acid (-sulfinic acid), sulfoxide (-sulfoxide), sulfide (-sulfide), sulfenamide (-sulfenamide), sulfonamide (-sulfonamide), sulfinamide (-sulfinamide), (-thioic acid), (thiocarboxy-), (-dithioic acid), (dithiocarboxy-), (thiocyanate) (isothiocyanate) |
| (phosphine), (phosphoric acid), (thiophosphoric acid), (dithiophosphoric acid), (phosphonic acid), (trithiophosphoric acid), (cyanophosphonic acid), (dithiphosphonic acid), (trithiophosphonic acid), (phosphinic acid), (thiophosphinic acid), (dithiophosphinic acid), (phosphenic acid), (phosphorous acid), (trithiophosphorous acid), (phosphonous acid), (phosphinous acid), (phosphenous acid), (phosphorimidic acid), (phosphonamidic acid), (phosphonimidic acid), (phosphinimidic acid), (phosphine oxide), (phosphine sulfide), (phosphine imide), (phosphorane) |
| lactam (-lactam), phenyl group (phenyl-), benzoyl group (benzoyl-), benzyl group (benzyl-), (benzene), (benzaldehyde), (benzoic acid), (phenol) |
| (naphtalene), (pentalene), (indene), (azulene), (biphenylene), (acenaphthylene), (fluorene), (phenalene), (phenanthrene), (anthracene), (fluoranthene), (acephenanthrylene), (aceanthrylene), (pyrene), (chrysene), (triphenylene), (perylene), (pleiadene), (picene), (tetraphenylene), (rubicene), (coronene), (trinaphthylene), (ovalene), (pyranthylene), (naphtacene), (pentacene) |



| | |
|---|---|
| ChemText | text editor (a Java application) |
| input method engine | |
| AdvIM | main class for the input method |
| AdvIMD | initializes the input method |
| CreateFormula | transforms molecular names into the molecular formula |
| CreateMolecule | creates various molecular formats (using CDK) |
| CreateRings | creates molecular data for ring compounds (using CDK) |
| M2View | displays 2D structures |

Figure 3: The classes implemented in this work

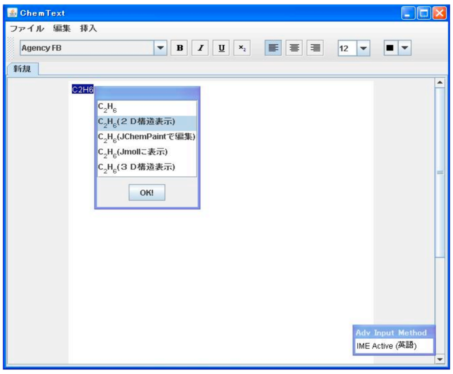


Figure 4: The snapshot of ChemIME

The whole paths of the data stream are shown in Fig.5. When the user inputs a compound name and presses the space key on the text editor, the input method system is invoked. In this stage, AdvIM object catches the event and get the string, then pass it to the CreateFormula object. This object analyzes the compound name and creates the molecular formula in return. AdvIM receives the formula and displays it on the editor. At the same time, if the user selects the item of "2D or 3D structure view" on the conversion list, then CreateFormula sends the molecular information to CreateMolecule or CreateRings objects to prepare generating 2D or 3D structures. CreateMolecule is used if the molecule does not form a ring structure. Otherwise CreateRings is used for the ring structure. These classes use the Molecule class which is a part of CDK. At this stage, several molecular input data (MDL mol, CML, Gaussian input format) which basically consist of molecular coordinates are created with standard interatomic bond information and passed to the external programs (JChemPaint [15] or Jmol [16]) by M2View object.

As has been shown previously, although in its prototype stage, the present system could convert many molecular names into molecular formulas, 2D, and 3D structures without preparing each dictionary data.

The input method has still serious errors, for example:

- Several pentavalent phosphorous compounds and polycyclic compounds are unavailable to display 3D structure view in the window.
- Java VM needs memory more than 120 MB to invoke the input method.

The former seems to be caused by the bugs of the preference file of CDK. The latter is because of huge size of template file of ring compounds of CDK. Therefore the modification of the source code and template file of CDK should be needed. These problems are now under consideration.

As future work, we plan to improve and extend our software such as:

- supporting other file formats
- improving to treat ring compounds
- error analysis for displaying some 3D structures

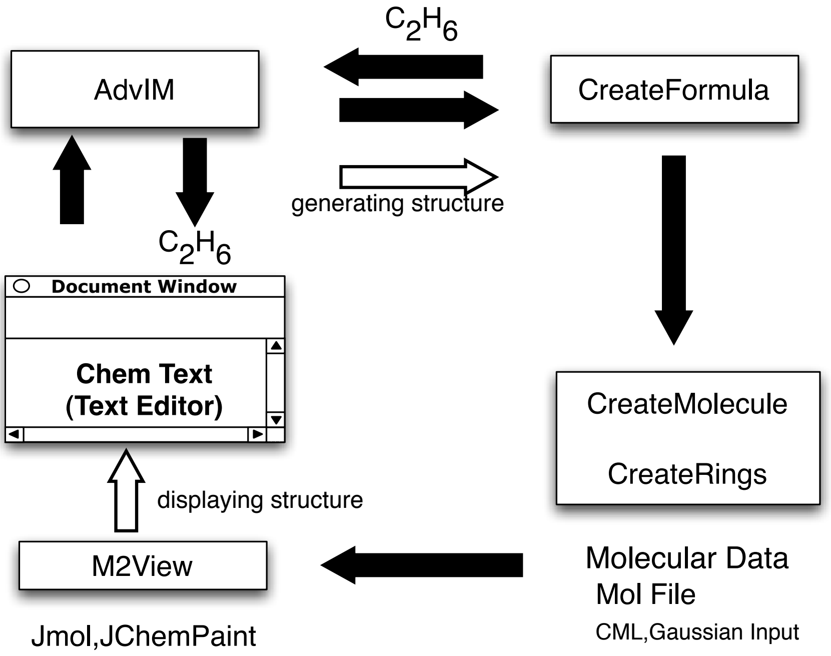


Figure 5: The whole paths of the data stream after invoking ChemIME on ChemText

- capability of more complex IUPAC names

and also to be open to scientific community.

## 4 Conclusions

In this study we have designed and implemented an input method engine (ChemIME) that can convert basic IUPAC names into corresponding molecular formulas, 2D or 3D structures. We have also developed a text editor (ChemText) that can treat RTF format documents to display subscript number and 2D structures in the document. They are all written in Java, and using Java Input Method Framework and some open source libraries. However, they have still some problems in the code that should be modified.

### Acknowledgments

The authors thank Mr Suyama and Miss Usui for their help.

## References

[1] Nomenclature of Organic Chemistry, Sections A, B, C, D, E, F, and H, Pergamon Press, Oxford, 1979. Copyright 1979 IUPAC.

[2] A Guide to IUPAC Nomenclature of Organic Compounds (Recommendations 1993), 1993, Blackwell Scientific publications, Copyright 1993 IUPAC.

[3] ChemDraw, CambridgeSoft, USA.

[4] Chemistry 4D Draw, ChemInnovation Software, Inc., USA.

[5] K. Arnold, J. Gosling, D. Holmes, The Java Programming Language, 4th ed., Prentice Hall (2005).

[6] C. Steinbeck, Y. Han, S. Kuhn, O. Horlacher, E. Luttmann, E. Willighagen, The Chemistry Development Kit (CDK): An Open-Source Java Library for Chemo- and Bioinformatics, J. Chem. Inf. Comput. Sci., 43, 493-500 (2003).

[7] P. Murray-Rust and H. Rzepa, Chemical Markup, XML, and the Worldwide Web. 1. Basic Principles, J. Chem. Inf. Comput. Sci., 39, 928-942 (1999).

[8] A. Dalby, J. G. Nourse, W. D.Hounshell, A. K. I. Gushurst, D. L. Grier, B. A. Leland and J. Laufer, Description of Several Chemical Structure File Formats Used by Computer Programs Developed at Molecular Design Limited, J. Chem. Inf. Comput. Sci., 32, 244-255 (1992).

[9] The Protein Data Bank, http://www.rcsb.org/pdb/.

[10] D.Weininger, SMILES, a Chemical Language and Information System. 1. Introduction to Methodology and Encoding Rules., J. Chem. Inf. Comput. Sci., 29, 97-101 (1988).

[11] S.E.Stein, S. R. Heller and D. Tchekhovskoi, An Open Standard for Chemical Structure Representation: The IUPAC Chemical Identifier., In Proeedings of the 2003 International Chemical Information conference. Nimes, Franc, October 19-22; Infonortics: Tetury, UK, pp 131-143.

[12] Gaussian 03, Revision C.02, M. J. Frisch, G. W. Trucks, H. B. Schlegel, G. E. Scuseria, M. A. Robb, J. R. Cheeseman, J. A. Montgomery, Jr., T. Vreven, K. N. Kudin, J. C. Burant, J. M. Millam, S. S. Iyengar, J. Tomasi, V. Barone, B. Mennucci, M. Cossi, G. Scalmani, N. Rega, G. A. Petersson, H. Nakatsuji, M. Hada, M. Ehara, K. Toyota, R. Fukuda, J. Hasegawa, M. Ishida, T. Nakajima, Y. Honda, O. Kitao, H. Nakai, M. Klene, X. Li, J. E. Knox, H. P. Hratchian, J. B. Cross, V. Bakken, C. Adamo, J. Jaramillo, R. Gomperts, R. E. Stratmann, O. Yazyev, A. J. Austin, R. Cammi, C. Pomelli, J. W. Ochterski, P. Y. Ayala, K. Morokuma, G. A. Voth, P. Salvador, J. J. Dannenberg, V. G. Zakrzewski, S. Dapprich, A. D. Daniels, M. C. Strain, O. Farkas, D. K. Malick, A. D. Rabuck, K. Raghavachari, J. B. Foresman, J. V. Ortiz, Q. Cui, A. G. Baboul, S. Clifford, J. Cioslowski, B. B. Stefanov, G. Liu, A. Liashenko, P. Piskorz, I. Komaromi, R. L. Martin, D. J. Fox, T. Keith, M. A. Al-Laham, C. Y. Peng, A. Nanayakkara, M. Challacombe, P. M. W. Gill, B. Johnson, W. Chen, M. W. Wong, C. Gonzalez, and J. A. Pople, Gaussian, Inc., Wallingford CT, 2004.

[13] CDK, Chemistry Development Kit, http://sourceforge.net/projects/cdk/.

[14] iText, a Free Java-PDF Library, http://www.lowagie.com/iText/.

[15] JChemPaint, http://sourceforge.net/projects/jchempaint/.

[16] Jmol, http://sourceforge.net/projects/jmol/.