ARTICLES

Internet Software for the Calculation of the Lipophilicity and Aqueous Solubility of Chemical Compounds

Igor V. Tetko,*[,†,‡] Vsevolod Yu. Tanchuk,[‡] Tamara N. Kasheva,[‡] and Alessandro E. P. Villa[†]

Laboratoire de Neuro-Heuristique,[§] Institut de Physiologie, Université de Lausanne, Rue du Bugnon 7, Lausanne, CH-1005, Switzerland, and Biomedical Department, Institute of Bioorganic & Petroleum Chemistry, Murmanskaya 1, Kiev-660, 253660, Ukraine

Received July 3, 2000

In this paper we describe an Internet Java-based technology that allows scientists to make their analytical software available worldwide. The implementation of this technology is exemplified by programs for the calculation of the lipophilicity and water solubility of chemical compounds available at http://www.lnh.unil.ch/~itetko/logp. Both these molecular properties are key parameters in quantitative structure−activity relationship studies and are used to provide invaluable information for the overall understanding of the uptake distribution, biotransformation, and elimination of a wide variety of chemicals. The compounds can be analyzed in batch or single-compound mode. The single-compound analysis offers the possibility to compare our results with several popular lipophilicity calculation methods, including CLOGP, KOWWIN, and XLOGP. The chemical compounds are analyzed according to SMILES line notation that can be prepared with the JME molecular editor of Peter Ertl. Conversion to SMILES from 56 formats is also available using the molecular structure information interchange hub developed by Pat Walters and Matt Stahl.

INTRODUCTION

Internet activities have become in the last few years a major investment in information, business, communication, and teaching technologies and chemistry.[1−5] The WWW (World Wide Web) impact on society has dramatically increased, especially in the fields of education and scientific research. It is clear that the Internet will become a major system for knowledge extraction and education in the new century. A great deal of information is available for chemists in the form of chemical databases such as ChemFinder[6] and ChemExper Chemical Directory,[3] on-line journals and conferences, etc. Academic scientific research can have a specific place in this system by providing access to scientific programs developed by academia. Such programs developed by professionals can become available to a worldwide audience, thus providing applications across several disciplines of science and industry.

A large number of available scientific programs have been developed in Fortran and C/C++ programming languages. The question is how to make these software products publicly available through the Internet. The main idea is to do such integration as general as possible, flexible for extension of programs and incorporation of new modules with minimal changes in the existing software. This can be important to share scientific programs and methods of data analysis over the Internet.

In this study we introduce an Internet Java-based technology that makes it possible for scientists to make their analytical software available worldwide. The calculation of the lipophilicity and water solubility of chemical substances is an example of this technology. Both these molecular properties are key parameters in quantitative structure−activity relationship (QSAR) studies and are used to provide invaluable information for the overall understanding of the uptake distribution, biotransformation, and elimination of a wide variety of chemicals. These coefficients are very important in the process of drug discovery and the development from molecular design to pharmaceutical formulation and biopharmacy. The calculation of these parameters[7,8] using electrotopological state indices[9,10] and artificial neural networks is done using single-compound and batch modes. The single-compound mode provides a comparison of the calculated results with such popular methods as CLOGP,[11] KOWWIN,[12] and XLOGP.[13]

METHODS

The appearance of the Java language has dramatically influenced the WWW society.[14] Since the beginning, inspired by the first steps in the development of WWW browsers, Java creators envisioned the same Java program running on different types of computer chips and in many different operating environments. The Java interpreter is the key to running the same Java program working across different hardware platforms. The Java compiler does not convert a program to a machine language specific to the computer on

* To whom correspondence should be addressed at the Université de Lausanne. Fax: 41-21-692.5505. E-mail: itetko@eliot.unil.ch. World Wide Web: http://www.lnh.unil.ch/~itetko.
† Université de Lausanne.
‡ Institute of Bioorganic & Petroleum Chemistry.
§ World Wide Web: http://www.lnh.unil.ch.

10.1021/ci000393l CCC: $20.00
Published on Web 02/06/2001

which it will run. Instead, the compiler converts the program to machine language that runs on a theoretical machine, the so-called Java virtual machine (JVM). The JVM is implemented in software and represents the Java interpreter. The JVM is developed for most computer platforms and computer systems. Therefore, different Java interpreters allow the same Java program to run on different machines efficiently. The Internet packages of Java make it possible to run the programs across the Internet, to provide unprecedented possibilities for remote calculations, to search on large databases, and to offer various dynamic client−server interactions. Another important feature of the Java programs, called "applets", is that they can be easily accessible over the Internet using the popular WWW browsers, such as Netscape[15] and Internet Explorer.[16]

The Java programs can be integrated with C/C++ programs using the Java native interface (JNI).[17] An interface between Java and native code is programmed by declaring in a Java program native methods implemented using C/C++ code. Calls of such methods, i.e., C/C++ programs, allow a fast execution of time-critical code as well as reuse of the software C/C++ libraries previously developed by the users. The result of this call is directly parsed to the Java program. At the same time the JNI framework lets the native method utilize Java objects in the same way that Java code uses these objects. A native method can access objects created by Java code, create its own Java objects, and update them. The native method can call the existing Java method, pass it the required parameters, and get the results back when the method is completed. Thus, JNI provides a flexible method for both-way integration of C and Java languages, and therefore, it was used to develop the program presented in this study. Furthermore, programs written in Fortran could be converted to C/C++ programs.

The original log *P*/log *S* calculation software was programmed using C++ language and consisted of three separate programs, including one program to generate indices and two programs to calculate both molecular properties. The analyzed molecules were coded using line-notation code, SMILES.[18] The input parameters included extended E-state indices and molecular weight and were directly calculated from the molecular structures. The weights of artificial neural networks trained to predict molecular properties were saved on hard disk. A new program that incorporates the calculation of E-state indices and the prediction of lipophilicity and aqueous solubility (ALOGPS, an artificial neural network program for the calculation of log *P* and log *S*) using the previously stored neural network weights was created and integrated with Java using the JNI.

Despite the SMILES codes now becoming very popular, there are still many different chemical data formats used by chemists. The molecular structure information interchange hub, BABEL, developed by Pat Walters and Matt Stahl at the University of Arizona[19] offers a powerful conversion program for 56 chemical data formats. This software was integrated with Java to allow "on-fly" conversion of molecules in different data formats into SMILES code.

A user also has the possibility to prepare SMILES codes using the JME molecular editor developed by Peter Ertl (Novartis).[20] This editor was developed in Java and is analogous to editing molecules with standard drawing packages. The editor generates a SMILES code that is passed for further analysis to the server. This editor is a very convenient tool, especially for people who are not familiar with SMILES notation.

A number of different methods for the calculation of the lipophilicity of chemical compounds have been recently developed. It is possible that some methods could be more or less appropriate for specific types of compounds analyzed by the user. Thus, a user may be interested to compare the lipophilicity calculated using several different methods and then can decide which software should be used for his/her data series. Such a possibility exists in our program for the single-compound mode that displays the lipophilicity results calculated by our own model, ALOGPS, and three other programs, namely, CLOGP,[11] KOWWIN,[12] and XLOGP.[13] The demo versions of the first two programs are available on-line.[21] We have programmed an interface and made it possible to display the results of these programs at the same WWW page. The XLOGP program is freely distributed by the Institute of Physical Chemistry, Peking University.[22] This program was integrated with the Java interface in the same way as the BABEL and ALOGPS programs. The XLOGP program uses input data files in "Mol2" Sybyl[23] format and relies on the internal types of atoms generated by Sybyl. Sybyl uses several data formats to describe one atom, e.g., seven different types of nitrogen, N.1, N.2, N.3, N.4, N.ar, N.am, and N.pl3. The BABEL conversion program may produce incorrect conversion of atom types, especially for nitrogen. To overcome this problem, a program that provides a reliable conversion of SMILES codes to Sybyl Mol2 files was programmed.

RESULTS

The general layout of the developed software includes three main parts, namely, the client applet, the super server, and calculation servers. All three parts are important components of the developed software.

The client applet is the only part of the program that is visible to the user. It provides three ways for the user to submit his/her data.

First, the front page (Figure 1) contains an input field where the user can directly type or copy−paste the Chemical Abstracts Service registry number (CAS RN) of a compound or SMILES code. This is the main input to the program. When the "submit" button is pressed, the applet analyzes the input field, recognizes whether it is a SMILES code or a CAS RN, and performs syntax and checksum checking of the CAS RN.

The second way is provided by the "use JME" button. A click on this button opens a new HTML window with the JME molecular editor. The user can edit the molecule in this editor and submit it for calculation.

The third way includes the possibility to upload a local file directly from the end-user computer and to convert it on-fly to SMILES. This option is available through the "upload" button. A click on this button opens a window with the choice of 56 supported data formats available from the BABEL conversion program. However, a reliable conversion to SMILES codes can be provided only from nine data formats, namely, Alchemy, M3D, Macromodel, MDL MOL-file, Isis SDF, MM2, and Sybyl Mol and Mol2 files. The other data formats could sometimes produce incorrect

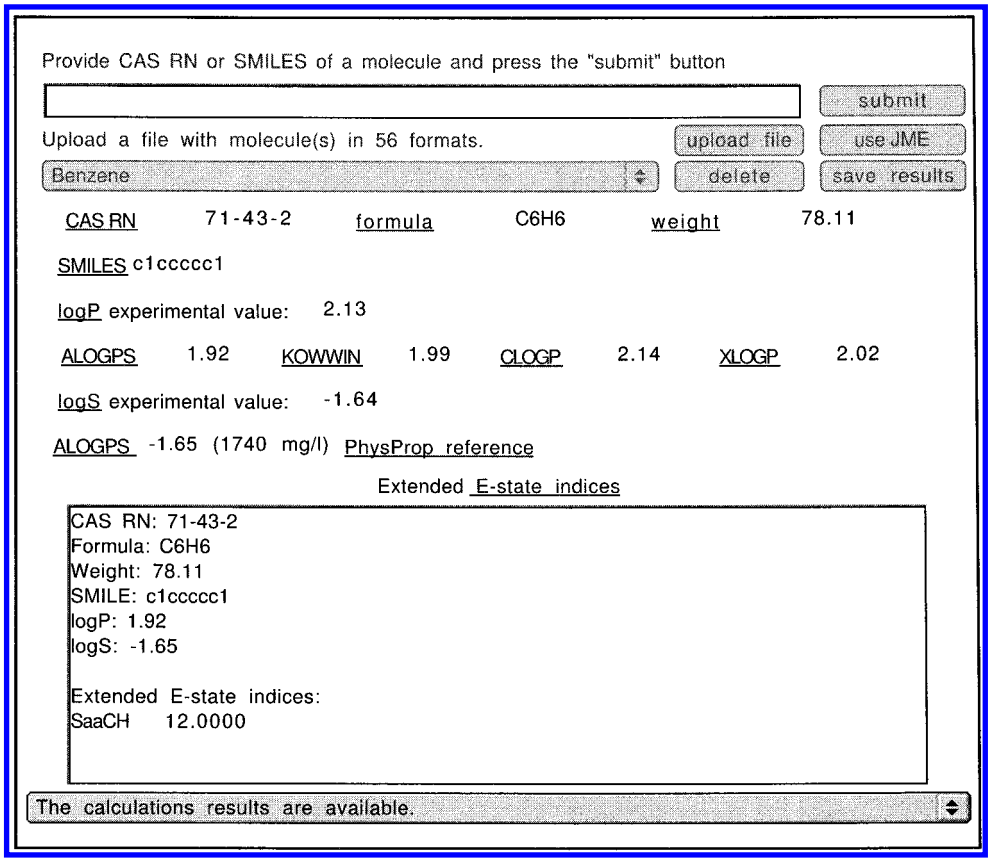

Figure 1. Front page of the WWW interface for lipophilicity/aqueous solubility calculations. A click on the underlined words or on the calculated molecular properties opens new Web pages with calculated results or information about the molecules.

conversion of some single/double bonds and aromatic atoms, especially if the aromatic ring contains heteroatoms. This problem cannot be simply improved since some data formats do not contain enough information about the types of atoms and/or their connectivity for unambiguous conversion of molecules. For this reason, the user must carefully check the resulting SMILES code and correct it if necessary. The converted compounds can be visualized by clicking the log *P* result calculated by CLOGP. The correction of SMILES can be done very fast and can save a lot of time required to convert the raw data files to SMILES. It is preferable that the users provide data as SMILES strings or prepare the molecule with the JME editor. The user can also use the BABEL conversion program available at our site.

Some data formats, for example, MDL Isis SDF, Sybyl files, etc., provide the possibility to store several molecules inside one file. In this case, all molecules are converted and analyzed in batch mode. The batch mode is also available if the user provides a file with several SMILES codes (one code per line).

The output results of the single-compound and batch modes are different. The single-compound mode can be used to compare results also calculated by the CLOGP,[11] KOWWIN,[12] and XLOGP[13] programs. A click on the label of each program opens a link to the home page of the corresponding program, while a click on the calculated log *P* value provides detailed information about its computation. More options are available for compounds with a known CAS RN. A click on "CAS RN" opens a ChemFinder window with the information about this specific compound. A similar link to the PhysProp database provides information about the physicochemical properties of the compound, including, if available, its lipophilicity and aqueous solubility. This page also provides experimental log *P* and log *S* values for the compounds. The molecules analyzed in single-compound mode can be stored locally on the user's disk by clicking the "save results" button. The user can specify which information should be saved. The saved E-state indices can be further used for QSAR studies.

The batch mode was created with the purpose of analyzing a large number of compounds. It is recognized automatically whenever two or more compounds are provided simultaneously. The output file calculated in this mode contains a table with calculated log *P*/log *S* values.

The client interface visualizes only the results, while their actual calculations are done using two important parts of the software that are hidden from the user. These are the super server and the calculation servers.

Super Server. The main purpose of the super server is to collect the requests from the client, to deliver the tasks to the calculation servers, and to deliver the calculated results to the client.

The super server provides a wide flexibility of the software. Its use eliminates Java security restrictions that forbid a client applet from establishing a connection with any server, except with the server from which it was uploaded.

The super server does not need to know which kind of data or tasks are transferred. This makes it possible to extend the tasks handled by the super server and to add new applications without recompilation or even stopping of the super server. In fact, in addition to the programs described

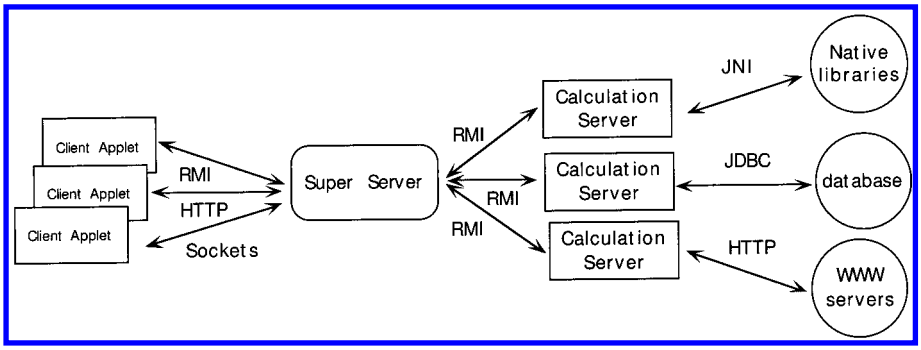

Figure 2. Protocols between the main components of the software: RMI, remote method invocation; JNI, Java native interface; JDBC, Java database connectivity interface; HTTP, hypertext transfer protocol.

in this study, two data analysis programs, namely, artificial neural networks and polynomial neural networks,[24] and several programs for data analysis in neurophysiology[25,26] are currently handled by the super server. The super server recognizes the applications according to an identification keyword, i.e., "logP" in the case of the log *P*/log *S* program or "ann" in the case of neural networks, and redirects the request to the corresponding calculation server. Both client applets and calculation servers register to the super server. The registration of an applet identifies the service required by the client, or alternatively the services provided by the calculation server. The super server builds up a list of available keywords during the registration of the calculation servers. New services can be easily added to this list with registration of new calculation servers. This makes it possible to extend the functionality of the software and to include new applications that can be developed by other companies or institutes.

Calculation servers provide conversions of data files, the calculation of the properties of molecules, and the search of stored molecules in a local database. These servers represent the third part of the software. One server can calculate several different tasks that are provided as native libraries. For example, the same server can be used to calculate lipophilicity and also to perform conversion of files. Each specific task has its own library corresponding to a specific keyword. The paths and names of the libraries are indicated in an external configuration file. The list of keywords is uploaded from an external configuration file of the calculation server and is provided to the super server during the registration process. The libraries are dynamically loaded when the calculation server receives a request for a specific task. The loaded libraries execute the task, and the results of the calculations are sent back to the client. It is important to mention that the original libraries may remain located at the private directory of the developer. Thus, they remain a property of the provider and are protected from unsanctioned access by clients or by anybody else.

The use of the technology presented here makes it possible to provide an optimal distribution of the computational power among the available computers. Several calculation servers can be installed on a multiprocessor server computer, thus allowing an optional use of its resources. The super server is also very useful to track calculation errors and to debug the software. The super server keeps in memory all tasks that are executed. If one task crashes, the data and configuration files can be retrieved for inspection and program debugging. Some tasks executed by a user could also require substantial time. In such a case the registered user does not need to wait in front of his/her browser and he/she can quit it. The submitted task will be calculated and will inform the user about its termination by e-mail. The user can login and collect the results even up to one week after the termination of the task. This option is available only for registered users (the super server should correctly recognize the owner of the task) and for tasks that require long calculation time, such as artificial neural network calculations. Since the calculation of the physical properties of compounds is done in several seconds, only on-line calculations are supported for the calculation of the physical properties of compounds. It should be noted that a similar possibility to upload calculated results after disconnection of a user is also a feature of the Advanced Chemistry Development (ACD) I-labs system[27] that was recently announced by this firm.[28]

The calculation server also performs a search of stored compounds according to their CAS RNs and SMILES codes in a local database. If the analyzed molecule is found in the local database of about 3000 compounds, the client applet will also display its name, its CAS RN, and, if available, its experimental values. If its CAS RN is not available in the local database, a search of the compound is performed in the CHEMEXPER database,[29] which includes about 75 000 compounds including SDF files with their structures. If the requested CAS RN is in the database, the program converts the SDF file using BABEL and calculates the properties of the compound.

Internet Protocols. The integration of all these programs was done with three specific Internet protocols (Figure 2), namely, HTTP (hypertext transfer protocol) requests, remote method invocation (RMI), and our custom protocol developed using Sockets.[14] The HTTP requests are generally executed when a user opens a new WWW page, e.g., by clicking a link in a browser. This simple event produces a cascade of complex operations by the browser and the WWW server to locate the requested page and to display it to the user. This protocol is a basic part of all Web applications.

The HTTP protocol is based on the transmission control protocol (TCP) that provides a reliable, point-to-point communication channel used by client−server applications to communicate between two computers on the Internet. The communication with the TCP functionally corresponds to communication using a fax machine or telegraph: one program (computer) calls the second program (computer) at a specific address, establishes a communication with it, and sends a message. The second program accepts the request for communication and reads the messages that were sent

by the first program. Like a fax machine, the TCP guarantees that data sent from one end of the connection actually get to the other end and in the same order they were sent. Otherwise, an error is reported. The HTTP, file transfer protocol (FTP), and terminal emulation program (Telnet) are all examples of applications that require a reliable communication channel. The order in which the data are sent and received over the network is critical to the success of these applications.

The communication using the TCP is done by way of Sockets, which represents one end point of a two-way communication link between two programs running on the network. The Sockets are bounded to some specific ports of the computer. The port numbers up to 1250 are reserved for internal use by system software, e.g., for communication using well-known protocols, such as the HTTP (port 80), RMI (port 1099), FTP (port 21), or Telnet (port 23).

Programs that are executed on different computers over the Internet represent a challenging software engineering task. Their programming requires taking into account different speeds of computers, delay of data transfer due to the low speed of Internet connection, security issues, different computer operation systems, etc. The combination of Java and RMI provides an excellent possibility to simplify such programming. The RMI application in general consists of two parts: a server and a client. A server application creates some remote objects and makes references to such objects available for the client. The client application gets the remote references and then invokes methods on them. RMI provides a mechanism by which the server and the client communicate and pass information back and forth. The essential property of RMI is that calls of the remote methods are executed exactly in the same way as calls of local methods. Another central and unique feature of RMI is the possibility to download and execute the code of an object's class if the class is not defined in the receiver's virtual machine. Thus, the types and the behavior of an object, previously available only in a single computer, is transmitted to another, possibly remote, computer. This allows new types to be introduced into a remote virtual machine, thus extending dynamically the behavior of an application. An extension of RMI, called the RMI over Internet inter-orb protocol (IIOP), delivers common object request broker architecture (CORBA) compliant distributed computing capabilities.[30] IIOP allows application components written in C++, Smalltalk, and other CORBA-supported languages to communicate with components running on the Java platform. Because of these powerful features, we used RMI to implement all interactions between the super server and the calculation servers. This allowed us an easy integration of different computers and operating systems into a single powerful system. For example, the super server is currently running on a Linux operation system, while the calculation servers are distributed over several computers including two different Sun operation systems, Linux, and sometimes Macintosh and Windows 95/98. The same Java code corresponding to the calculation servers is running on all these different computers. However, the native libraries were specifically compiled for each hardware. The use of different platforms requires a C/C++ and Java code absolutely compliant with language specifications and simplifies detection and correction of program errors that can not be easily discovered when using only one operation system.

The Java language provides a simple method for database management through the Java database connectivity (JDBC) interface.[31] This interface lets a user send structured query language (SQL) statements to a database and process the returned results. This allows uniform access to a wide range of relational databases. Access to a database is done using the open database connectivity (ODBC) interface. This C-based interface to SQL-based database engines provides a consistent interface for communicating with a database and for accessing database data. Individual vendors provide specific drivers or "bridges" to their particular database management system. The number of supported databases continuously increases, and there are more than 120 drivers available for users, including drivers for practically all commercial databases, such as Oracle, Sybase, and a number of free and shareware programs, such as MySQL[32] used in our project. The JDBC was used in our application to provide access to the database of compounds stored on a local computer.

As we mentioned above, we found RMI to be very useful for programming the communications between the super server and the calculation servers. Unfortunately this interface could not be used as the standard one for the super server and client applets, because it is not supported by Internet Explorer.[33] This browser does not include RMI libraries that are considered as optional and are available for free upload, e.g., from Microsoft[34] or from a third party.[35] The upload and installation of such libraries could be a complicated task for users. Thus, a custom protocol based on Sockets was implemented in our software for this browser.

DISCUSSION

We have developed Internet-based software for the calculation of the lipophilicity and aqueous solubility of chemical compounds. This software can be easily extended to include new programs for the calculation of the lipophilicity and aqueous solubility of chemicals as well as new computational methods to be used in chemistry. We hope to extend this software by including new methods for calculations of new parameters of molecules.

The ideas and methods used to develop this software can be useful to scientists who would like to make their methods of data analysis publicly available to other researchers. This can be very important especially for the studies involving complex methods of data analysis, such as artificial neural networks. The results calculated by these methods depend on seed values, the selection of training and test sets, learning algorithms, and/or specialized software and therefore cannot be easily reproduced. We believe that public access to such results is important to really estimate and validate the performance of these methods.

The Internet provides another specific protocol, the common gateway interface (CGI),[1] that can be used to develop a user-friendly interface similar to that we have described in this paper. The CGI represents an extension of HTTP requests, and it executes programs at the server side as a response to the user request. The results of the calculations are then shown to a user. The CGI is used to calculate lipophilicity in CLOGP and KOWWIN. There are, however, some inconveniences of this interface. Generally, every time

the user needs to do something serious, the CGI programs are required to recontact the server and to submit forms. This can result in substantial delays before the user can receive the calculated results. In the case of the Java applet, the code of the program is downloaded only once and new calculation results can be added without need to download the code again. For example, the developed software displays results calculated by KOWWIN and CLOGP programs. The results of the ALOGPS algorithm are always available to the user faster than the results of CLOGP and KOWWIN. This is due to the delays associated with an HTML request to a corresponding server and processing its results. The use of Java makes it possible to rapidly update the calculated results without a need to reload complete HTML pages. If CLOGP and KOWWIN programs are not available, no results will be shown for these programs. The use of the CGI could have substantial difficulties organizing such an interface. One possibility could be to wait for the results of all programs for some fixed time (let us say after *t* s) and to provide the final page with all results available for that time. However, if time *t* is too small, it is possible that not all results will be provided to the user. On the contrary, if time *t* is very large, the user will have to wait a very long time before getting the results from a remote server that is overloaded at the moment. The Java user can submit a new request without waiting for the results of the first one. All calculated results can be scrolled in the Java page and then saved on the local disk.

The provider who would like to have distributed computing will still have to develop intranet software probably using the same Java language or C/C++. This will require writing a complex interface program to read/write data prepared in specific formats and to integrate different software products and can require more time and effort than to develop such an interface in Java. The integration of programs executed by different operation systems and database management can even represent the most difficult engineering part of the problem.

As was mentioned in the Methods, one of the reasons to create the super server was to overcome the restrictions of Java by allowing the applet to establish a connection only with the server it was uploaded from. This problem could be addressed using a signed applet too. Such an applet can be allowed to make connection to other servers, read/write files from the user disk, or access local devices, such as a printer. However, there are some disadvantages of this technology. First, the signing and verification was only partly developed in Java 1.1 technology, and that is why there are no Java classes yet supported with the main Web browsers, i.e., Netscape and Internet Explorer.[36] This support can be added by installing the corresponding Java Plug-In.[37] In our personal experience, the users are not motivated to spend their time for the installation of additional software and to search and change configuration parameters. As an alternative approach it is possible to run the signed applet using Netscape or Internet Explorer proprietary packages. Unfortunately, both browsers require different developments, and in addition this service is not free. Second, the signed applet can receive total control over the file system of a user and thus, in some cases, can also access some private information on the user computer. It is not clear whether the user will be willing to grant such access to software. Third, even if a signed applet will be allowed to contact different servers, it can be very inefficient to organize multistep calculations that send data back and forth to several different servers. A super server connected to the calculation servers by high-speed intranet connections provides much faster calculation of data compared to sending and receiving data over the Internet. For example, at the University of Lausanne the speed of intranet connection with RMI is 300−500 kB/s compared to 1−50 kB/s for the Internet connections.

The problems associated with the configuration and running of the signed applet will partially disappear with the release of a new version of browsers supporting Java 1.2 technology. The permission/policy model coming provides a unified method to grant different permissions of Java applets, and thus such applets can become frequent citizens of the Internet cyberspace.

Another problem is that the WWW servers that handle many HTTP requests should not be heavily loaded. This is usually the case if such servers perform only WWW services. On the contrary, the analysis of the large amount of data by calculation servers could require substantial computational time and resources. This can significantly slow the work of the WWW server and can even provoke its crash. To use a WWW server as a relay server only substantially decreases its loading.

The three-tier architecture, i.e., the architecture that includes a bottleneck server to access system resources such as calculation servers or databases, etc., is becoming widespread recently, especially to organize efficient access to databases of the Internet users.[38−40] Three-tier Java-based software for data collection of HIV/AIDS patients was reported as a flexible and extensive tool handling different kinds of digitized data.[38,39] A new system provided a considerable performance improvement over the HTML/CGI protocol previously used for these purposes. The authors also addressed a number of security issues and used Java encryption features to provide a secure transfer of patient information over the Internet. A part of the critical Java code, to ensure high security, was distributed to the users by postal mail. A use of signed applets and public/private key now provides an alternative way to distribute such information over the Internet. The advantages of the three-tier system, such as reusability, flexibility, significant reduction in costs and effort of development, and the possibility of migration to appearing technologies, are also discussed for the clinical information collection system in modern health care.[40]

The Java technology is becoming an important part of commercial software. Examples of such software include the WWW-based chemical information system developed at Novartis[20,41] and the Interactive Laboratory of ACD Inc.[42] The ACD company uses Java-based Web software to provide predictions of various physicochemical molecular properties including boiling points, log *P*/*D*, aqueous solubility, and others parameters as well as other database services in different fields of chemistry.

In summary, we have developed Internet software for the calculation of the lipophilicity and aqueous solubility of chemical compounds and described the main components of this software. Our package is open to include new programs developed by other scientists who can contact us to receive sample code and instructions on how to implement this software. This can provide a worldwide dissemination

of academic results and will have a positive impact on chemical research.

ACKNOWLEDGMENT

This study was partially supported by Grants INTAS-Ukraine 95-0060 and INTAS-OPEN 97-0168. We are grateful to Luhua Lai, Gao Ying, and Renxiao Wang (Peking University) for providing us the source code of their program and for fruitful feedback for the development of the SMILES to Mol2 conversion program. We thank Luc Patiny (University of Lausanne) for the possibility of accessing CHEMEXPER. Peter Ertl (Novartis) is greatly acknowledged for providing the JME molecular editor. We thank David Livingstone, Javier Iglesias, Dmitry Filipov, and Jarmo Huuskonen for their suggestions and participation in the development of some software modules and Val Kulkov (ACD) for his useful remarks.

REFERENCES AND NOTES

(1) Wiggins, G. Chemistry on the Internet—the Library on Your Computer. *J. Chem. Inf. Comput. Sci*. **1998**, *38*, 956−965.
(2) Heller, S. R. Chemistry on the Internet—the Road to Everywhere and Nowhere. *J. Chem. Inf. Comput. Sci*. **1998**, *38*, 956−965.
(3) Patiny, L. Sharing Product Physical Characteristics over the Internet. *Int. J. Chem.* **2000**, *3*, 1−6.
(4) Tonge, A. P.; Rzepa, H. S.; Yoshida, H. Authentication of Internet-based Distributed Computing Resources in Chemistry. *J. Chem. Inf. Comput. Sci*. **1999**, *39*, 483−490.
(5) Rzepa, H. S.; Tonge, A. P. VChemLab: A Virtual Chemistry Laboratory. The Storage, Retrieval and Display of Chemical Information using Standard Internet Tools. *J. Chem. Inf. Comput. Sci*. **1998**, *38*, 1048−1053.
(6) Brecher, J. S. The ChemFinder WebServer: Indexing Chemical Data on the Internet. *Chimia* **1998**, *52*, 658−663. http://www.chemfinder-.com/.
(7) Huuskonen, J. J.; Livingstone, D. J.; Tetko, I. V. Neural Network Modeling for Estimation of Partition Coefficient Based on Atom-Type Electrotopological State Indices. *J. Chem. Inf. Comput. Sci*. **2000**, *40*, 947−955.
(8) Tetko, I. V.; Tanchuk, V. Yu.; Kasheva, T. N.; Villa, A. E. P. Estimation of Aqueous Solubility of Chemical Compounds using E-state Indices. *J. Chem. Inf. Comput. Sci*., submitted for publication.
(9) Kier, L. B.; Hall, L. H. An Electrotopological State Index for Atoms in Molecules. *Pharm. Res.* **1990**, *7*, 801−807.
(10) Hall, L. H.; Kier, L. B. Electrotopological State Indices for Atom Types: A Novel Combination of Electronic, Topological, and Valence State Information. *J. Chem. Inf. Comput. Sci*. **1995**, *35*, 1039−1045.
(11) Leo, A. Calculating logPoct from Structures. *Chem. Rev*. **1993**, *93*, 1281−1306.
(12) Meylan, W. M.; Howard, P. H. Atom/Fragment Contribution Method for Estimating Octanol−Water Partition Coefficients. *J. Pharm. Sci*. **1995**, *84*, 83−92.
(13) Wang, R.; Gao, Y.; Lai, L. Calculating Partition Coefficient by Atom-Additive Method. *Perspect. Drug Des.* **2000**, *19*, 47−66.
(14) Chan, P. *Java Developers Almanac*; Addison-Wesley: Reading, MA, 1999.
(15) http://www.netscape.com/.
(16) http://www.microsoft.com/windows/ie/.
(17) Gordon, R. *Essential JNI: Java Native Interface*, 1st ed.; Prentice-Hall: Upper Saddle River, NJ, 1998.
(18) Weininger, D. SMILES 1. Introduction and Encoding Rules. *J. Chem. Inf. Comput. Sci*. **1988**, *28*, 31. See also http://www.daylight.com/dayhtml/smiles/.
(19) A source code of this program is available at http://smog.com/chem/babel/.
(20) Ertl, P.; Jacob, O. WWW-based Chemical Information System. *J. Mol. Struct.* **1997**, *419*, 113−120.
(21) CLOGP is available at http://www.daylight.com/daycgi/clogp and KOWWIN at http://esc-plaza.syrres.com/interkow/kowdemo.htm.
(22) XLOGP v2.0 is available by anonymous FTP to ftp2.ipc.pku.edu.cn, directory "pub/software/xlogp".
(23) Sybyl is a software package of Tripos, Inc., http://www.tripos.com.
(24) Tetko, I. V.; Aksenova, T. I.; Volkovich, V. V.; Kasheva, T. N.; Filipov, D. V.; Welsh, W. J.; Livingstone, D. J.; Villa, A. E. P. Polynomial Neural Network for Linear and Nonlinear Model Selection in Quantitative-Structure Activity Relationship Studies on the Internet. *SAR QSAR Environ. Res.* **2000**, *11*, 263−280.
(25) Villa, A. E. P.; Tetko, I. V.; Iglesias, J.; Filipov, D. V.; Transdisciplinary Approach To Scientific Data Analysis Through Internet. *Proceedings of the International Transdisciplinary Conference*, Zurich, Feb 27 to Mar 1, 2000; pp 550−555.
(26) Villa, A. E. P.; Tetko, I. V.; Iglesias, J. Computer Assisted Neurophysiological Analysis of Cell Assemblies Activity. *Neurocomputing*, submitted for publication.
(27) http://www.acdlabs.com/ilab/.
(28) http://www2.acdlabs.com/ilab/info/whatsnew.html.
(29) http://www.chemexper.com.
(30) http://java.sun.com/products/rmi-iiop/.
(31) http://java.sun.com/products/jdbc/.
(32) http://www.mysql.com/.
(33) http://www.java.sun.com/lawsuit/.
(34) http://www.microsoft.com/Java/resource/misc.htm.
(35) http://alphaworks.ibm.com.
(36) http://java.sun.com/security/signExample/.
(37) http://www.java.sun.com/products/plugin/.
(38) Sippel, H.; Eich, H. P.; Ohmann, C. Data Collection in Multi-Center Clinical Trials via Internet. A Generic System in Java. *Medinfo* **1998**, *9*, 93−97.
(39) Sippel, H.; Ohmann, C. A Web-Based Data Collection System for Clinical Studies Using Java. *Med. Informatics* **1998**, *23*, 223−229.
(40) Chu, S.; Cesnik, B. A Three-Tier Clinical Information Systems Design Model. *Int. J. Med. Informatics* **2000**, *57*, 91−107.
(41) Ertl, P. QSAR Analysis through the World Wide Web. *Chimia* **1998**, *52*, 673−677.
(42) Williams, A. The Advanced Chemistry Development Toolset and the Interactive Laboratory, ACD/Ilab. *ChemNews.Com* **1999**, *9*, 4.

CI000393L