Internet Accessible Compilers for Education in Software and Computer Engineering

Aleksander Malinowski
Dept. of ECE
Bradley University
Peoria, IL 61625
e-mail: olekmali@ieee.org

Bogdan M. Wilamowski
Dept. of Electrical Engineering
University of Wyoming
Laramie, WY 82071
e-mail: wilam@ieee.org

KEYWORDS

Network computing; Software engineering; Compilers;

ABSTRACT

This paper demonstrates the capability of educational Intranet and Internet use for software and computer engineering. There are several benefits that make networked software desirable. A web-based application can be used remotely throughout any network connection. Any operating system can be used to access it, making it independent of the operating system. Also an application can be made to run on a pay-per-user basis if licensing is desired, and much less installation time and configuration time is required, because the software is located on one central machine.

Internet Compilers Package (ICP) is an example of a networked software-engineering tool. Different compiler packages can be used for various stages of application development providing either maximum error detection or code optimization. A quick general introduction to the problem of Intranet and Internet programming is also provided as background information necessary to understand the currently emerging new technology.

INTRODUCTION

During the process of software development frequently more than one compiler package is required. Some products are known to be very useful for locating errors or debugging, while others perform extremely well when a program or library is in the final stage of development and should be optimized as much as possible. Also when facing obscure error messages, which may result in a time-consuming search for the error, a different error message from the second compiler frequently cuts that time dramatically. Having found it handy to have a few compilers ready for use the authors decided to apply this idea also for educational purposes.

Three compilers: GNU, Microsoft, and Inprise (formerly Borland) have been used in the university intranet. Web-page based front end allows to access them without any restrictions regarding the computer system requirements thus allowing for their use on different operating system platforms and on older machines with lesser performance. Adding a cross-compilation option to compilers capable of doing so solves the problem of binary code execution on multiple platforms. Access to selected commercial software components is enabled based on the user's computer IP address and a password. In that way students can always use the best tools at least for error detection no matter what computer they use. They can also access some additional features such as source code reformatting or compilation into assembly language of a particular microprocessor. The latter feature turned out to be very helpful in teaching computer architecture courses.

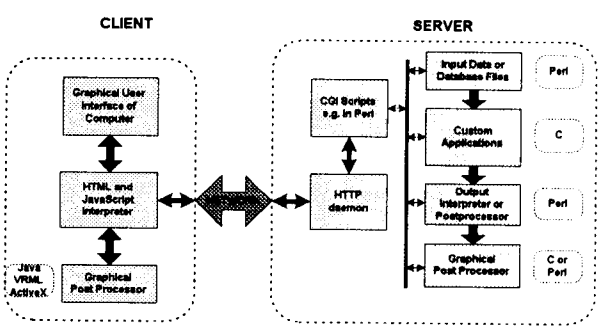

Fig. 1. Data flow in a network-based application.

INTRANET COMPILERS ARCHITECTURE

During software development it is important to justify which part of the software should run on the client machine and which part should run on the server. Applets are transferred through network when requested and execution is performed entirely on the client machine that made the request. With CGI much

less information has to be passed to the server. The server executes instructions based on the given information and sends the results back to the local machine that made the request. Fig. 1 shows the program component division and data flow in a network based application.

Network programming uses distributed resources. Part of the computation is done on the server and another part on the client machine. Certain information must be frequently sent both ways between client and server. It would be nice to follow the JAVA applet concept and have most of the computations done on the client machine. This approach, however, is not always feasible for three major reasons:

- Some programs are very large and thus not practical for sending entirely via network as applets.
- Software developers are giving away their software without the ability of controlling its usage.
- JAVA applets used on-line and on demand are slower than regular software.

The Internet bandwidth is already adequate for many applications if their data flow is carefully designed. Furthermore the bandwidth limitation will significantly improve with time. It is therefore important to develop methods, which take advantage of networks and then platform independent browsers. The web-based programming requires solving several issues such as:

- Minimization of the amount of data which must be sent by a network
- Task partitioning between the server and client
- Development of special user interfaces
- Security and account handling
- Portability of software used on servers and clients

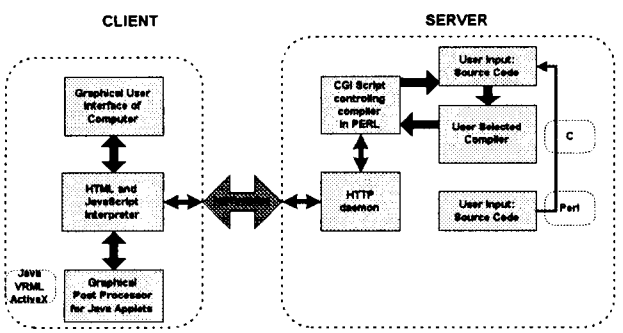

Fig. 2. Data flow in an Internet Compilers package.

These software design problems were solved in the ICP. The user interface is programmed in HTML enhanced with JavaScript. A PERL script handles the data flow, which is shown in Fig. 2. That script does the file managing, runs compilers and processes the compilation results. The result is both the source code listing and a binary code to download or a list of errors sent back to the user.

To use ICP, paste the program code from your compiler text editor, or from any text editor, to the web page form. Then submit the form. The compilation will be performed by PERL script on the server in batch mode. Although the front end is designed to be as simple as possible with only a few commonly used options, it is sufficiently functional and can be used quickly. The PERL script located on the server has to deal with the translation of these common options to the actual options of compilers from different vendors. It also handles the compilation errors and processes the report. The cookie technology, together with JavaScript, is used to store each user's preferences for compilation so that the settings are preserved when the user returns to the ICP web page.

INTRANET COMPILERS USER INTERFACE

Currently the Intranet Compilers package supports C, C++, Pascal, Fortran, and Java. It utilizes GNU, Borland, and Microsoft compilers for C and C++, GNU compiler for Fortran and Pascal, and Sun's JDK for Java. A common front end is used for all compilers (Fig. 3). This HTML page allows for selecting a vendor and a language, and for setting a few basic compilation options.

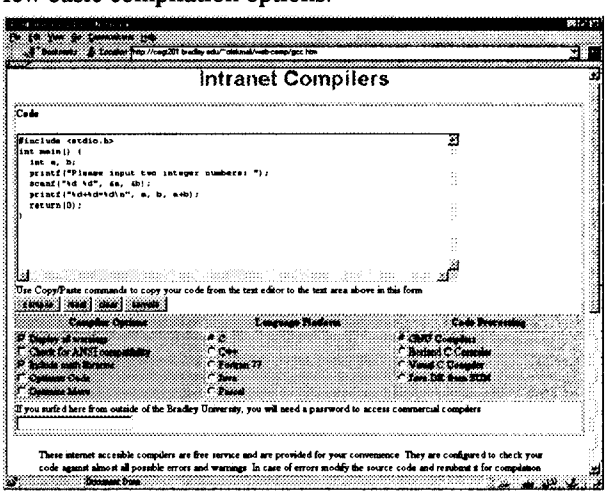

Fig. 3. A common front-end to all compilers.

The process of compilation is performed in batch mode. After setting the desired options and pasting the source code into the appropriate text box, the task can be started by pressing the ***COMPILE*** button. As a result another web page with HTML wrapped

information is sent back to the user. The result page is displayed in another browser window so that the user can correct the source code in the original window and resubmit it if necessary. Fig. 4 shows the sample compilation results for program, which contains errors. One of the major advantages of consolidating more than one compiler is the ability to cross-reference error message among different vendor products used from the same interface.

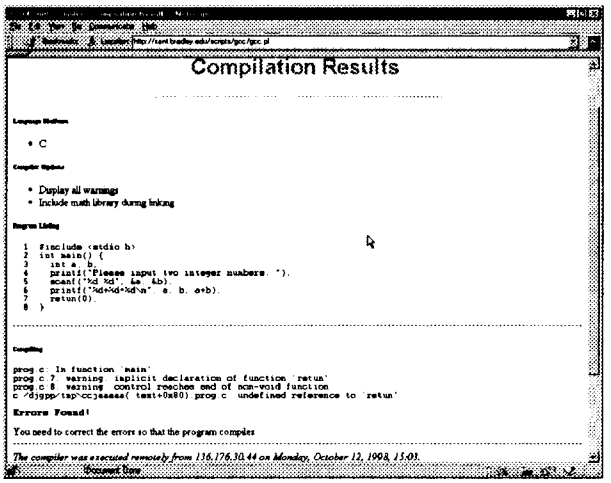

Fig. 4. Intranet Compilers results for errorous program.

Fig. 5 demonstrates the successful compilation message. In that case the compiled binary code is available for download. If the user's operating system agrees with the destination platform set for the compiler the binary code may be executed. At this moment it is not possible to execute the code on the server due to limited security of the server's operating system.

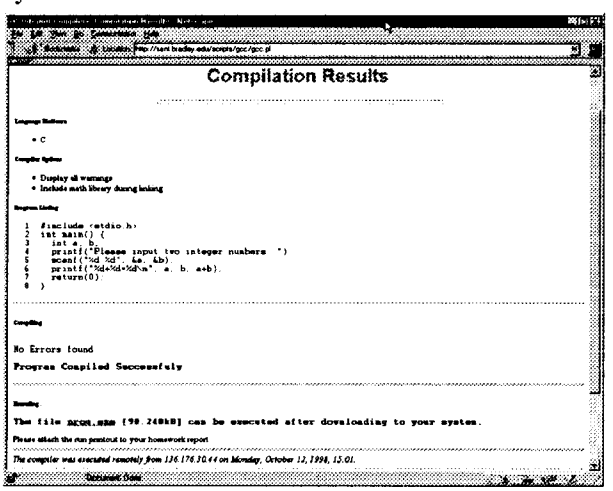

Fig. 5. Intranet Compilers results for successful compilation. The binary code can be downloaded.

Because of additional requirements on Java applets, an additional specialized front end is created for that language. If the compilation into bytecode is successful, the user can inspect the applet immediately. Fig. 6 shows the language specific front end.

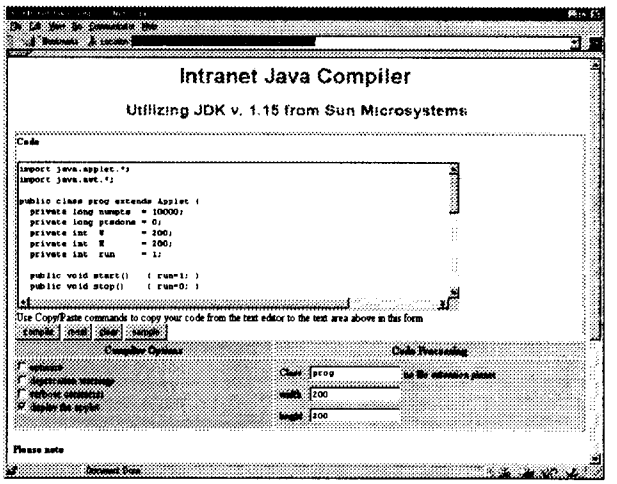

Fig. 6. Specialized front end for Java compiler.

Another specialized page was created for the partial compilation into assembly language, because of different and limited options used for this task. The assembly front-end and the compilation results are shown in Fig. 7. At this moment only 16- and 32-bit Intel assembly languages are available. However, a cross-compiler may be used to produce assembly language code for different machines.

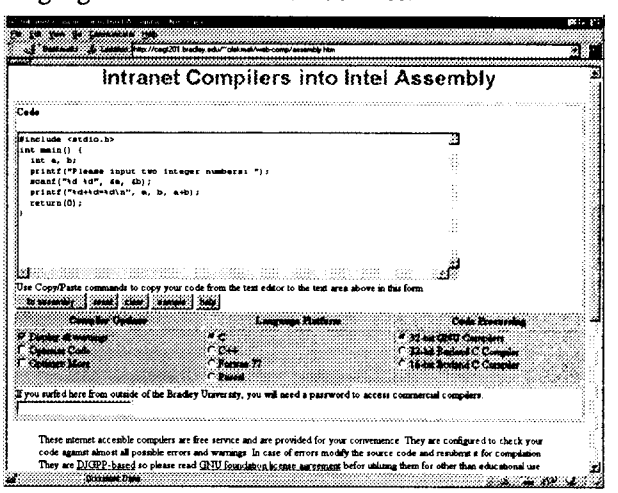

Fig. 7. Specialized front end for assembly language generator.

Another service available at ICP is automatic code formatting performed according to the user's specification. An HTML front end was developed for

GNU source code beautifier. This approach allows even beginner to use the program, which is rather difficult to configure otherwise. Fig. 8 shows the front end. The program is very useful in the first programming course, when beginner programmers try to experiment what coding style fits them the best. It also quite useful in case of parentheses mismatch errors. Missing parentheses can be more easily detected by the automatic indentation inspection especially in programs with many nested loops.

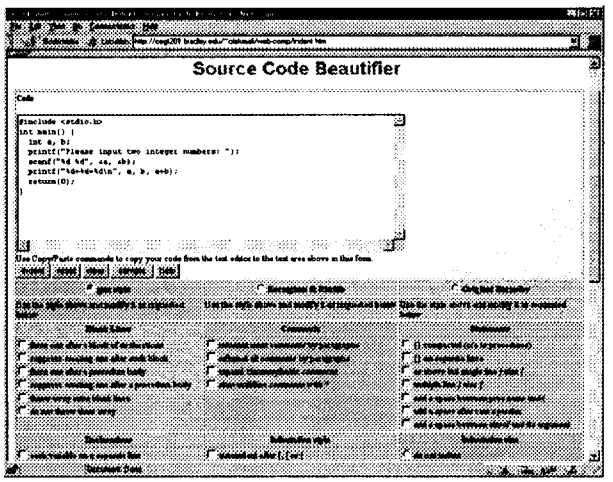

Fig. 8. Source code beautifier configuration page.

Everybody can inspect and utilize the Intranet Compilers Package by visiting the web site located at http://sant.bradley.edu/web-comp/. When one wants to use ICP over the Internet, only the limited version of the package is accessible. For obvious reasons, everybody can access only the compilers that do not require a license. Commercial compilers (Borland and Microsoft) are accessible only in the Intranet environment for users with selected IP addresses for which the license is bought.

CONCLUSION

The paper shows how to use the new opportunity created by Internet technologies for the efficient and platform independent usage of software engineering tools. The presented ICP are just an example, but it shows a way in which the technology can be implemented. The possibility to compare error messages from different compilers is very useful especially for students who do not have much experience in coding and debugging yet. The ability to use different compilers allows proficient programmer to pick up the fastest or the most convenient tool to compile the code and remove the errors , and then to use the product which allows for the best code optimization for the compilation of the final version of the program. Furthermore the availability of design tools via the Internet will boost design processes in many new communities, small businesses, and improve our education processes at universities by allowing students to use the same sophisticated software as is used by leading industries.

REFERENCES

[1] Sweet W., Geppert L., "http:// It has changed everything, especially our engineering thinking", *IEEE Spectrum*, January 1997, pp. 23-37.

[2] Jamsa K., Lalani S., Weakley S., *Web Programming,* Jamsa Press, Las Vegas, NV, 1996.