CLIPS: A Tool for the Development and Delivery of Expert Systems

Gary Riley
Software Technology Branch
NASA Johnson Space Center
Mail Stop PT4
Houston, TX 77058

ABSTRACT

The 'C' Language Integrated Production System (CLIPS) is a forward chaining rule-based language developed by the Software Technology Branch at the Johnson Space Center. CLIPS provides a complete environment for the construction of rule-based expert systems. CLIPS was designed specifically to provide high portability, low cost, and easy integration with external systems. Other key features of CLIPS include a powerful rule syntax, an interactive development environment, high performance, extensibility, a verification/validation tool, extensive documentation, and source code availability. The current release of CLIPS, version 4.3, is being used by over 2,500 users throughout the public and private community including: all NASA sites and branches of the military, numerous federal bureaus, government contractors, 140 universities, and many companies.

INTRODUCTION

Expert system technology is a major subset of Artificial Intelligence and has been aggressively pursued by researchers since the early 1970's. In the last few years, both government and commercial application developers have given expert systems considerable attention as well. An entire industry has grown to support the development of expert system tools and applications, with a wide variety of both hardware and software products now available. The availability of expert system tools has greatly reduced the effort and cost involved in developing an expert system.

Despite all this, expert systems have generally failed to make a major impact in application environments. This failure has stemmed from tool vendor's overemphasis on expert system development environments to the detriment of options for delivery of expert systems and training in expert system technology. Viable delivery options are necessary to field expert systems. Training options in expert system technology are necessary for the widest possible dissemination of this technology.

The 'C' Language Integrated Production System (CLIPS) is a forward chaining rule-based production system developed by the Software Technology Branch at NASA/Johnson Space Center. Version 1.0 of CLIPS, developed in the spring of 1985 in a little over two months, accomplished two major goals. The first of these goals was to gain useful insight and knowledge about the construction of expert system tools and to lay the groundwork for future versions. The second of these goals was to address the delivery problems of integrating and embedding expert systems into conventional environments. Version 1.0 successfully demonstrated the feasibility of continuing the project.

Subsequent development of CLIPS greatly improved its portability, performance, and functionality. A reference manual [1], architecture manual [2], and user's guide [3] were written. The first release of CLIPS, version 3.0, was in July of 1986. The latest version of CLIPS, version 4.3, was completed in June of 1989. A version of CLIPS developed entirely in Ada and fully syntax compatible with the C version of CLIPS has also been developed. CLIPS is currently available through COSMIC (see appendix).

DELIVERY

CLIPS was designed, in part, to solve problems related to the delivery of expert systems. CLIPS addresses several issues key to the delivery problem. Among these issues are: the ability to run on conventional hardware; the ability to run on a wide variety of hardware platforms; the ability to be integrated with and embedded within conventional software; low-cost delivery options; the ability to separate the development environment from the delivery environment (i.e. run-time modules); the ability to run efficiently (both speed and memory), and migration paths from development to delivery environments.

One major requirement for a delivery tool is the ability to run on conventional hardware. Portability of the expert system tool code insures the ability to deliver on a wide range of hardware from microcomputers to minicomputers to mainframes. Because CLIPS is written in C and special care was taken to preserve portability, CLIPS is able to provide expert system technology on a wide variety of conventional computers. CLIPS has been hosted on over a dozen brands of computer systems ranging from microcomputers to mainframes without code changes. To maintain portability, CLIPS utilizes the concept of a portable kernel. The kernel represents a section of code which utilizes no machine dependent features. To provide machine dependent features, such as windowed interfaces or graphics editors, CLIPS provides fully documented software hooks which allow machine dependent features to be integrated with the kernel.

At the time of its development, CLIPS was one of the few tools that was written in C and capable of running on a wide variety of conventional platforms. Most state-of-the-art expert system software tools at that time were based in LISP and ran only on specialized LISP hardware, such as the Symbolics or TI Explorer. In recent years, many tool vendors have migrated their products to workstations and PCs and numerous products are now available in C, Ada, and other conventional languages.

The ability to integrate with and embed within existing code is an important feature for a delivery tool. Integration guarantees that an expert system does not have to be relegated to performing tasks better left to conventional procedural languages. It also allows existing conventional code to be utilized. The capability to be embedded allows an expert system to be called as a subroutine (representing perhaps only one small part of a much larger program). Many tools view themselves as the "master" program and only permit control to be passed to other programs through them. CLIPS allows integration with C programs as well as integration with other languages such as FORTRAN and ADA. In addition, many functions are provided which allow CLIPS to be manipulated externally. Because the source code is available, CLIPS can be modified or tailored to meet a specific user's needs.

Applications should be delivered as economically as possible. Many tools require the entire development environment to run an application. This necessitates buying a new copy of the tool for every delivered application. Some tools provide the capability to generate run-time modules. These run-time modules are basically equivalent to the executable modules generated by compilers for procedural languages. Run-time modules allow the unneeded functionality and information associated with the development environment to be stripped away from the delivery environment. This is a desirable characteristic, but for many tools, each copy of a run-time module must be purchased.

CLIPS effectively addresses the problems of low cost delivery. The cost for CLIPS source code is $250. This initial cost provides unlimited copies of CLIPS for delivery, development, and training. In addition, CLIPS also provides the capability to generate run-time modules.

Another key feature for a delivery tool is efficiency. CLIPS is based on the Rete algorithm [4] which is an extremely efficient algorithm for pattern matching. CLIPS version 4.3 compares quite favorably to other commercially available expert system tools based on the Rete algorithm.

CURRENT USES

Although CLIPS was originally developed to help in the construction of aerospace related expert systems, it has been put to widespread usage in a number of fields. The current release of CLIPS, version 4.3, is being used by over 2,500 users throughout the public and private community including: all NASA sites and branches of the military, numerous federal bureaus, government contractors, 140 universities, and many companies. At the First CLIPS Conference held in August 1990, over 80 papers were presented on a diverse range of topics. In addition to aerospace and engineering applications, some other examples of CLIPS applications include: software engineering [5, 6], networking [7, 8], medical and biological [9, 10], and agricultural [11, 12].

EXTENSIONS

One of the key appeals of the CLIPS language results from the availability of the approximately 25,000 lines of CLIPS source code. Because the development of an expert system tool can require many man-years, the benefits of using CLIPS as a starting point for research and the creation of special purpose expert system tools cannot be understated. CLIPS users have enjoyed a great deal of success in adding their own extensions to CLIPS due to the source code availability and its open architecture. Some of the many extensions added by CLIPS users include: an SQL interface [13, 14], neural network extensions [15, 16], blackboard extensions [17], parallel and distributed processing [18, 19, 20, 21, 22], and object-oriented programming extensions [23, 24, 25].

Because CLIPS was written to be portable, its standard interface is a simple, text-oriented, command prompt. However, CLIPS provides the capability to layer an interface on top of CLIPS to provide the advantages of a windowed, mouse-driven, menu-oriented user interface. Currently, two such interfaces are provided with CLIPS: one for the Apple Macintosh family of computers and one for IBM PC compatible computers. An X-Windows interface is currently under development. Many users have also developed their own interfaces and interface extensions [26, 27, 28, 29, 30].

FUTURE ENHANCEMENTS

CLIPS is a continually evolving product. The next release of CLIPS, version 5.0, is scheduled for release in January 1990 [31]. The primary addition to CLIPS 5.0 will be the CLIPS Object-Oriented Language (COOL). The major capabilities of COOL are: class definitions with multiple inheritance and no restrictions on the number, types, or cardinality of slots; message passing which allows procedural code bundled with an object to be executed; and query functions which allow groups of instances to be examined and manipulated. In addition to COOL, numerous other enhancements have been added to CLIPS including: generic functions (which allow different pieces of procedural code to be executed depending upon the types or classes of the arguments), integer and double precision data type support, multiple conflict resolution strategies, global variables, logical dependencies, type checking on facts, full ANSI compiler support, and incremental reset for rules.

CONCLUSION

Because of its portability, extensibility, capabilities, and low-cost, CLIPS has received widespread acceptance throughout the government, industry, and academia. The development of CLIPS has helped to improve the ability to deliver expert system technology throughout the public and private sectors for a wide range of applications and diverse computing environments.

REFERENCES

1. Culbert, C., *CLIPS Reference Manual*. NASA document JSC-22948, July 1989.

2. Riley, G., *CLIPS Architecture Manual*. NASA document JSC-23047, May 1989.

3. Giarratano, J., *CLIPS User's Guide*. NASA document, August 1989.

4. Forgy, C. "Rete: A Fast Algorithm for the Many Pattern/Many Object Pattern Match Problem". Pages 17-37, Artificial Intelligence 19, (1982).

5. Cameron, B., and Helbig, H. "Hardware Independence Checkout Software", Proceedings of the First CLIPS Conference, Houston, TX., August 1990.

6. Morris, K. "Automating Symbolic Analysis with CLIPS", Proceedings of the First CLIPS Conference, Houston, TX., August 1990.

7. Hansen, R, and Flores, L. "JESNET Expert Assistant", Proceedings of the First CLIPS Conference, Houston, TX., August 1990.

8. Wannemacher, T. "Using CLIPS in a Distributed System: The Network Control Center (NCC) Expert System", Proceedings of the First CLIPS Conference, Houston, TX., August 1990.

9. Batra, S, and Macinnes, M. "GENEX: A Knowledge-Based Expert Assistant for Genbank Data Analysis", Proceedings of the First CLIPS Conference, Houston, TX., August 1990.

10. Salzman, G., Duque, R., Braylan, R., and Stewart, C. "A CLIPS Expert System for Clinical Flow Cytometry Data Analysis", Proceedings of the First CLIPS Conference, Houston, TX., August 1990.

11. Engel, B., Jones, D., Rhykerd, R., Rhykerd, L., Rhykerd Jr., C., and Rhykerd, C. "A CLIPS Expert System for Maximizing Alfalfa (Medicago Sativa L.) Production, Proceedings of the First CLIPS Conference, Houston, TX., August 1990.

12. Thai, C., Pease, J., and Shewfelt, R. "A Decision Support System for Delivering Optimal Quality Peach & Tomato", Proceedings of the First CLIPS Conference, Houston, TX., August 1990.

13. Snyder, J., and Chirica, Laurian. "An SQL Generator for CLIPS", Proceedings of the First CLIPS Conference, Houston, TX., August 1990.

14. Wess, B. "ΣCLIPSE = Presentation Management + NASA CLIPS + SQL", Proceedings of the First CLIPS Conference, Houston, TX., August 1990.

15. Charnock, E., and Eng, N. "CLIPS on the NeXT Computer", Proceedings of the First CLIPS Conference, Houston, TX., August 1990.

16. Bhatnagar, H., Krolak, P., McGee, B., and Coleman, J. "A Neural Network Simulation Package in CLIPS", Proceedings of the First CLIPS Conference, Houston, TX., August 1990.

17. Orchard, R., and Diaz, A. "BB_CLIPS: Blackboard Extensions to CLIPS", Proceedings of the First CLIPS Conference, Houston, TX., August 1990.

18. Geyer, S. "CLIPS Meets the Connection Machine or How to Create a Parallel Production System", Proceedings of the First CLIPS Conference, Houston, TX., August 1990.

19. Goldstein, D. "PRAIS: Distributed, Real-Time Knowledge-Based Systems Made Easy", Proceedings of the First CLIPS Conference, Houston, TX., August 1990.

20. Myers, L, Johnson, C., and Johnson, D. "MARBLE – A System for Executing Expert Systems in Parallel", Proceedings of the First CLIPS Conference, Houston, TX., August 1990.

21. Taylor, J., and Myers, L. "Executing CLIPS Expert Systems in a Distributed Environment", Proceedings of the First CLIPS Conference, Houston, TX., August 1990.

22. McGee, B., Miller, M., Krolak, P., and Barr, S. "Integrating PCLIPS into ULowell's Lincoln Logs Factory of the Future", Proceedings of the First CLIPS Conference, Houston, TX., August 1990.

23. Logie, D., and Kamil, H. "Integration of Object-Oriented Knowledge Representation with the CLIPS Rule Based System", Proceedings of the First CLIPS Conference, Houston, TX., August 1990.

24. Sobkowicz, C. "An Object Oriented Extension to CLIPS", Proceedings of the First CLIPS Conference, Houston, TX., August 1990.

25. Assal, H, and Myers, L. "Implementation of a Frame-Based Representation in CLIPS", Proceedings of the First CLIPS Conference, Houston, TX., August 1990.

26. Engel, B., Rewerts, C., Srinivasan, R., Rogers, J., and Jones, D. "CLIPS Interface Development Tools and Their Application", Proceedings of the First CLIPS Conference, Houston, TX., August 1990.

27. Pickering, B., and Hill, R. "HyperCLIPS: A HyperCard Interface to CLIPS", Proceedings of the First CLIPS Conference, Houston, TX., August 1990.

28. Callegari, A. "Integrating Commercial off-the-shelf (COTS) Graphics and Extended Memory packages with CLIPS", Proceedings of the First CLIPS Conference, Houston, TX., August 1990.

29. Faul, B. "Constructing Complex Graphics Applications With CLIPS and the X Window System", Proceedings of the First CLIPS Conference, Houston, TX., August 1990.

30. Feagin, T. "A Graphical Interface to CLIPS Using Sunview", Proceedings of the First CLIPS Conference, Houston, TX., August 1990.

31. Riley, G., and Donnell, B. "Advanced CLIPS Capabilities", Proceedings of SOAR '90, Albuquerque, NM., June 1990. Houston

APPENDIX

CLIPS is free to NASA, USAF, and their contractors for use on NASA and USAF projects by calling the CLIPS Help Desk between the hours of 9:00 AM to 4:00 PM (CST) Monday through Friday at (713) 280-2233. Government contractors should have their contract monitor call the CLIPS Help desk to obtain CLIPS. Others may obtain CLIPS through the Computer Software Management and Information Center (COSMIC), which is the distribution point for NASA software. The program number is COS-10022. The program price is $250.00, and the documentation price is $62.00 (as of July 1990). The program price is for the source code. Price discounts are available to U.S. academic institutions. Further information can be obtained from

COSMIC
382 E. Broad St.
Athens, GA 30602
(404) 542-3265