McESE-FranzLISP: McMASTER EXPERT SYSTEM EXTENSION OF FranzLISP

F. Franek[1]

Dept. of Comp. Sci. and Systems
McMaster University
Hamilton, Ontario
L8S 4K1 Canada

ABSTRACT.

McESE - McMaster Expert System Environment is a software tool designed to help create problem-specific shells with imprecise and incomplete knowledge, and fast and compact expert systems applications in a particular programming language. This is achieved by "extending" the programming language with a set of commands allowing communication with McESE knowledge bases and McESE special software (see [FB], or [FB1]). In this paper implementation details of FranzLISP extension (called McESE-FranzLISP) are presented and discussed.

(1) INTRODUCTION.

As described in [FB] and [FB1], McESE is a programming environment for building of expert systems. Its main features include the ability to specify the way of handling uncertainty for the whole knowledge base, or for each rule separately, to utilize compiled rule-form knowledge bases for fast inferring, and the ability to build the expert system in a particular programming language. The software of McESE is written in the programming language C for two reasons; compact code that leads to fast execution, and the ability to program in C low-level tasks efficiently enough. In this paper the extension of FranzLISP is described and discussed. From now on we may use term lisp instead of FranzLISP for the sake of brevity.

The task of extending FranzLISP to McESE-FranzLISP consisted of two major tasks:

- to make the software of McESE written in C an integral part of the lisp system with sufficient communication between McESE software and lisp system without disturbing the lisp system too much and without a need of extensive re-writing of the existing McESE software. It must be ensured that any of the C functions of McESE required can be invoked from within lisp, and conversely, any lisp function, be it built-in or user defined, interpreted or compiled, can be invoked from within the C functions;

[1] Research supported by SERB 5-26397 and NSERC OGP0025112 research grants.

- to make the non-lisp data structure of compiled knowledge base (we shall call it knowledge tree in accordance with [FB] and [FB1]) an integral part of lisp system accessible from within lisp, but protected from lisp's garbage collector as long as needed.

Of course, there were some other problems to be resolved as well, e.g. how to simplify invocation of McESE, how to protect the components of McESE from the user and so on.

(2) HOW TO MAKE McESE SOFTWARE AN INTEGRAL PART OF FranzLISP.

Since the kernel of FranzLISP in UNIX is written in C, the major lisp function eval can be directly invoked from any C program loaded into the lisp system. This fact was used to solve the problem of invocation of arbitrary lisp functions from within C functions. This is achieved by creating a special list in FranzLISP system, passing it to the C function which modifies it and than `eval` is invoked with the special list passed to it. This kind of communication allows the full range necessary and does not intrude upon lisp at all. Given the fact that FranzLISP's kernel is written in C, had more technical and implementation details of FranzLISP been available, we could have modified our C code to extend the original kernel directly. This approach did not seem appropriate for the actual implementations on different machines may differ, and for too much of intrusion to the system. The means we opted for instead do slow the execution a bit (which is not of such a crucial importance given the slow execution of lisp in general) but allow for extension of lisp independent from the implementation details of FranzLISP's kernel.

(3) HOW TO MAKE McESE KNOWLEDGE BASE AN INTEGRAL PART OF FranzLISP.

When compiled knowledge tree is to loaded to main memory, it must stay in the lisp system as long as the user desires while some C and lisp functions are executed. Since the garbage collector of FranzLISP is invoked automatically when the system has not enough memory available, the knowledge tree must be protected against it. It must also be protected against intrusion by any other function of the system. On top of it, when the user does not need the knowledge tree any longer, the memory it takes must be made available to the system again. These requirements were satisfied by "storing" the knowledge tree within a lisp data structure, so-called immediate vector. Since immediate vectors carry binary data, there is no problem with the "contents" being the knowledge tree. As long as this immediate vector is bound, the systems considers it active and hence it is protected from the garbage collector as well as all lisp functions. On the other hand, when the knowledge tree is unloaded, its immediate vector is unbound, and so the immediate vector is not considered active by the system and hence it is accessible to garbage collector. From the point of view of the inference engine and other components of McESE accessing the knowledge tree, all they need to know is the beginning of the tree and correct address links within the tree. Thus the fact that the knowledge tree is inside a lisp object matters not to them.

(4) CONCLUSION.

The McESE-FranzLISP is the FranzLISP extension by the McESE system. It provides almost full range of McESE capabilities as described in [FB], [FB1], moreover, McESE-FranzLISP provides more built-in cvpf's. The system is being currently used in the fourth year CS course at McMaster University Architecture of Expert Systems, and its graduate version. The system satisfies its goals well and can be a valuable addition to the repertoire of knowledge engineers, mainly for the ability to make fast prototypes in McESE-FranzLISP with a low overhead costs before the expert system is built in a faster and more compact language (as in McESE-C) utilizing the same McESE knowledge bases as the prototype in McESE-FranzLISP.

REFERENCES.

[F] F. Franek, McESE-FranzLISP: *McESE Expert System Extension of FranzLISP*, Tech. Rep., Dept. of Comp. Sci. and Systems, McMaster University, Hamilton, Ont., Canada, 1988.

[FB] F. Franek, I. Bruha, *McESE - NcMaster Expert System Environment*, to appear in Proc. of ICCI'89, Volume 1, North-Holland, 1989.

[FB1] F. Franek, I. Bruha, *The McESE project*, Tech. Rep., Dept. of Comp. Sci. and Systems, McMaster University, Hamilton, Ont., Canada, 1988.