

Standard-Output

**Guy L. Steele Jr.
Thinking Machines Corporation**

Allow me to introduce myself: I am vice chair of ANSI Technical Committee X3J13 (Common Lisp). I have been working for almost nine years on Common Lisp, along with many dozens of other people who have contributed to the definition, development, and implementation of Common Lisp.

The effort began innocently enough in 1981, when several research and implementation groups decided to join forces to curb the divergence of their respective dialects. I volunteered to draft and edit the language specification, and three years later the first edition of *Common Lisp: The Language* was published by Digital Press. The stated goals of the language design were commonality among certain existing dialects, portability, consistency, expressiveness, compatibility, efficiency, power, and stability (in roughly that order). (Note that ease of implementation and ease of learning were not among these goals! The general intent was to design a language by power implementors for power users.)

Through what I believe was a fortunate combination of luck, timing, and technical excellence, Common Lisp caught on. Since publication of the first specification in 1984, many implementors have used it as a *de facto* standard for Lisp implementation. As a result, it is now much easier to port large Lisp programs from one implementation to another. Common Lisp has proved to be a useful and stable platform for rapid prototyping and systems delivery in artificial intelligence and other areas.

The 1984 definition of Common Lisp was imperfect and incomplete. In some cases this was inadvertent: some odd boundary situation was overlooked and its consequences not specified, or different passages in were conflict, or some property of Lisp was so well-known and traditionally relied upon that I forgot to write it down. In other cases the informal committee that was defining Common Lisp could not settle on a solution, and therefore agreed to leave some important aspect of the language unspecified rather than choose a less than satisfactory definition. An example is error handling; 1984 Common Lisp had plenty of ways to signal errors but no way for a program to trap or process them.

Over the next year I collected reports of errors in the book and gaps in the language. In December 1985, a group of implementors and users met in Boston to discuss the state of Common Lisp. I prepared two lists for this meeting, one of errata and clarifications that I thought would be relatively uncontroversial (boy, was I wrong!) and one of more substantial changes I thought should be considered and perhaps voted upon. Others also brought proposals to discuss. It became clear to everyone that there was now enough interest in Common Lisp, and dependence on its stability, that a more formal mechanism was needed for managing changes to the language.

This realization led to the formation of X3J13, a subcommittee of ANSI committee X3, to produce a formal American National Standard for Common Lisp. After a great deal of debate, X3J13 eventually agreed on the following set of goals for itself:

1. X3J13 is chartered to produce an American National Standard for Common Lisp. It will codify existing practice, provide extensions to facilitate portability of code among diverse implementations, and establish normative Common Lisp programming practice.
2. The committee will begin with the language described in *Common Lisp: The Language* by Guy L. Steele Jr. (Digital Press, 1984), which is the current *de facto* standard for Common Lisp. Whenever there is a proposal for the standard to differ from *Common Lisp: The Language*, the

committee shall weigh both future costs of adopting (or not adopting) a change and costs of conversion of existing code. Aesthetic criteria shall be a subordinate consideration.

3. The committee will address at least the following topics in the course of producing the standard, in each case either incorporating specific features or explaining why no action was taken:
 - (a) Repairing mistakes, ambiguities, and minor omissions in *Common Lisp: The Language*
 - (b) Error handling and condition signalling
 - (c) Semantics of compilation
 - (d) Object-oriented programming
 - (e) Iteration constructs
 - (f) Multiprocessing
 - (g) Graphics
 - (h) Windows
 - (i) Validation
 - (j) One versus two namespaces for functions and variables

Topics (a)-(c) concern deficiencies in *Common Lisp: The Language* that require resolution. Topics (d) and (e) are not addressed by *Common Lisp: The Language* but appear to be well-understood and ready for standardization. Topics (f)-(i) concern areas where standardization is desirable but not crucial to production of a standard. Topic (j) is an area of current controversy within the Lisp community. Other topics may be considered if specific proposals are received.

4. The committee recognizes that Lisp programming practice will continue to evolve and anticipates the need for future revisions and extensions to the standard. This may include a family of Lisps and/or a layered Lisp model.
5. X3J13 is committed to work with ISO toward an international Lisp standard.

It is worthwhile to review here, point by point, the actual progress made toward these goals.

6. X3J13 has indeed completed the bulk of its technical work in rectifying the 1984 definition and codifying extensions to that definition that have received widespread use and approval. The committee agreed by a formal vote in June 1989 that, while additional correction and polishing will be required, the committee foresees no further major revisions to the language for the first version of the standard.
7. The committee did begin with the 1984 edition of *Common Lisp: The Language*, but decided that its style, format, and lack of rigor was unsuitable for standardization purposes. Therefore the actual draft standard has been recast into a format contributed by Lucid, Inc., and rewritten by Kathy Chapman of Digital Equipment Corporation with assistance from the X3J13 Drafting Subcommittee. The content of *Common Lisp: The Language* has continued to serve as the yardstick against which to compare all proposed language changes. The last sentence, concerning aesthetic criteria, was worded very carefully after a great deal of debate, and is a two-edged sword. It represents a compromise between those who care foremost about elegance and cleanliness in language design and those who don't want matters of taste to impede effective solutions to practical problems. While aesthetic considerations shall be subordinate criteria, they shall be criteria that may legitimately be raised in support of certain design choices.
8. (a) The committee has made dozens and dozens of corrections to the 1984 language specification. These range from trivial-but-it-ought-to-be-fixed to blatant-inconsistency to major-hole-in-the-language.

An example of the trivial: where a `FORMAT` specifier has both a colon and an `atsign` modifier, the 1984 specification did not specify whether they must appear in a particular order or may appear in either order. All the examples happened to put the colon first as a matter of consistent style. X3J13 has decreed that the user may write them in either order.

An example of an inconsistency: two different passages in the 1984 specification gave conflicting definitions of the size of a hash table. X3J13 has chosen one of these as definitive.

An example of a hole: the 1984 specification made a few vague references to the fact that certain stream operations may be applied to closed streams. X3J13 voted to specify clearly and explicitly a list of these operations and their precise behaviors when applied to closed streams.

The Cleanup Subcommittee, chaired by Larry Masinter, has managed the flow of proposals in this category.

(b) An elaborate error-handling and condition-signalling mechanism has been added to the language, based on a specification by Kent Pitman.

(c) The Compiler Subcommittee, chaired by Sandra Loosemore, has tied down many loose ends in the areas of compilation, semantics of compiled code, and interaction between compiler and interpreter. These include the treatment of constants in compiled code, the compile-time processing of top-level forms in a file, and the manner in which compilation errors and warning are reported.

(d) The Common Lisp Object System, parts 1 and 2, has been accepted for inclusion in Common Lisp. The CLOS specification was written by Daniel Bobrow, Linda DeMichiel, Richard Gabriel, Sonya Keene, Gregor Kiczales, and David Moon, and has been published in *SIGPLAN Notices* (September 1988) and in *Lisp and Symbolic Computation* (January 1989).

X3J13 showed great interest in part 3 (the Metaobject Protocol), but concurred with the authors that this part of CLOS probably could not be agreed upon and put into a form suitable for standardization within the desired time frame. X3J13 has taken great care to keep open the possibility of including the Metaobject Protocol in some future revision of Common Lisp.

(e) The Iteration Subcommittee, chaired by Jon L. White, examined three different proposals for new iteration constructs in Common Lisp: `LOOP`, `series`, and `generators`. The first has a keyword-based syntax; the other two are closely related and have a functional syntax. By way of example, suppose that we wish to make an association list giving the integers from 1 to 10 and their square roots. Using `LOOP` one might write

```
(loop for x from 1 to 10
      collect (cons x (sqrt x)))
whereas using series one might write
(let ((x (scan-range 1 :upto 10)))
  (collect-alist x (sqrt x)))
```

X3J13 voted to include `LOOP` in the forthcoming draft standard. X3J13 expressed interest in `series` and `generators`, but the consensus as of January 1989 was that these other approaches were not yet sufficiently mature or in sufficiently widespread use to warrant inclusion in the draft Common Lisp standard at that time. However, the subcommittee was directed to continue work on these approaches and X3J13 is open to the possibility of standardizing them at a later date.

(f) X3J13 has received no proposals for standardizing on any aspect of multiprocessing. In my judgement this is still a subject of intense research. Many different ideas are being pursued, such as futures, process queues, and parallel data structures; it would be premature to attempt standardization at this time.

(g) X3J13 is making no attempt to define a standard graphics interface. It has examined a number of existing graphics toolkits, but none seems appropriate for inclusion in a standard at this time.

(h) X3J13 is making no attempt to define a standard windows interface. While the X Window System may be of considerable interest, it came along late in the game. X3J13 has decided not to delay the production of the Common Lisp standard at this time. It is possible that a set of standard bindings for the X interface, or some other window system, may be standardized in the future.

(i) X3J13 early on made some investigations into validation methodologies and the possibility of defining a suite of validation tests, but has now declined to specify an official validation methodology or test suite for Common Lisp.

(j) The matter of one versus two namespaces for functions and variables was hotly debated, with good arguments on both sides. Eventually it was decided to retain the dual namespaces of the 1984 specification. The namespace distinction therefore remains one of the more conspicuous differences between Common Lisp and Scheme.

(k) There was no item (k) in the original set of goals, but this seems a good place to mention three other major changes to the language. The Character Subcommittee, chaired by Thom Linden, suggested a number of changes to the character-handling facilities that will facilitate international use of Common Lisp. Richard Waters designed an interface to the pretty printer that allows the user to specify how user-defined data structures are to be pretty-printed and arranges for `FORMAT` to interact correctly with the pretty printer. Finally, the file system interface has been cleaned up and extended to facilitate even greater portability.

X3J13 remains open to the notion of a layered language definition, and has discussed the possibility with its ISO counterpart. However, X3J13 has declined to create a layered definition or to specify formal subsets for the purposes of an ANSI standard. (There was considerable discussion on the question of whether to make CLOS an optional module. Eventually it was decided to make CLOS mandatory so that it might be better integrated with the rest of the language.)

X3J13 has consistently sent one or more representatives to ISO meetings. They have represented not only the Common Lisp community but also the interests of the Scheme community at the international level.

The production of a standard is a painfully slow process. While X3J13 believes that it has made all major technical decisions necessary for a first version of a standard, undoubtedly numerous minor problems remain.

A draft standard is now being prepared. My guess is that it will be available in late 1990. There will then be a period (required by ANSI) for public review. X3J13 must then consider the comments it receives and respond appropriately. If the comments result in substantial changes to the draft standard, multiple public review periods may be required before the draft can be approved as an American National Standard.

If you have an interest in the forthcoming Common Lisp standard, stay tuned. The best thing you can do is to review the draft standard when it comes out; I'll announce it in this column (among other places). If we are to produce a standard that truly reflects the needs of many users and implementors, there must be timely review and feedback. When the time comes, let X3J13 know what you think!

THE WORLD ACCORDING TO SIR BLOOPER

It is not infrequent that one reads an otherwise well-written piece of prose to find abuse of the English language. The would-be author should be ever vigilant to the possibility of abuse. Here is a splendid, albeit juvenile, example of what I mean. Its source is not known to me. I've copied the text as I received it from an anonymous donor.

The World According to Student Bloopers

=====

One of the fringe benefits of being an English or History teacher is receiving the occasional jewel of a student blooper in an essay. I have pasted together the following "history" of the world from certifiably genuine student bloopers collected by teachers throughout the United States, from eighth grade through college level. Read carefully, and you will learn a lot.

The inhabitants of ancient Egypt were called mummies. They lived in the Sarah Dessert and traveled by Camelot. The climate of the Sarah is such that the inhabitants have to live elsewhere, so certain areas of the dessert are cultivated by irritation. The Egyptians built the Pyramids in the shape of a huge triangular cube. The pramids are a range of mountains between France and Spain.

The Bible is full of interesting caricatures. In the first book of the Bible, Guinnesses, Adam and Eve were created from an apple tree. One of their children, Cain, once asked, "Am I my brother's son?" God asked Abraham to sacrifice Isaac on Mount Montezuma. Jacob, son of Isaac, stole his brother's birth mark. Jacob was a patriarch who brought up his twelve sons to be patriarchs, but they did not take to it. One of Jacob's sons, Joseph, gave refuse to the Israelites.

Pharaoh forced the Hebrew slaves to make bread without straw. Moses led them to the Red Sea, where they made unleavened bread, which is bread made without any ingredients. Afterwards, Moses went up on Mount Cyanide to get the ten commandments. David was a Hebrew king skilled at playing the liar. He fought with the Philatelists, a race of people who lived in Biblical times. Solomon, one of David's sons, had 500 wives and 500 porcupines.

Without the Greeks we wouldn't have history. The Greeks invented three kinds of columns-Corinthian, Doric, and Ironic. They also had myths. A myth is a female moth. One myth says that the mother of Achilles dipped him in the River Stynx until he became intollerable. Achilles appeared in "The Iliad", by Homer. Homer also wrote "The Oddity", in which Penelope was the last hardship that Ulysses endured on his journey. Actually, Homer was not written by Homer but by another man of that name.

Socrates was a famous Greek teacher who went around giving people advice. They killed him. Socrates died from an overdose of wedlock.