プログラミング言語　6－5
（１９８６．５．２３）

Common Lisp 言語仕様の現在の論点とその方向

井田昌之（青山学院大学）

概要　Common Lisp の言語仕様は1984年に出版された。それ以後も着実に、主に電子メールによる討議、半年に一度程度のミーティングにより、仕様の相互理解、不明確・難解な記述の解明が少しずつ進められ、多くの人に期待に答えるべき仕様となってきた。また、多くの処理系が出現するようになった。現在、ANSIおよびISO でCommon Lisp をベースとしたLispの標準化のための委員会が提案され、具体的な活動を開始する時期に至った。ここでは、ここに至る過程とその方向について、その歴史的な位置づけ、付加・拡充の対象として議論のある機能などを、現在の時点での分析に基づいて述べる。

"Common Lisp Approach to the Lisp Standardization and its current issues"
(in japanese)

by Masayuki Ida
Aoyama Gakuin University,Morinosato-aoyama 1,Atsugi, Kanagawa, Japan 243-01

Abstract:　The specification of Common Lisp was published in 1984. After the year, along with the electronic mail communications and the roughly once a half year meeting, mutual understanding of the spceification and the clarification of the notation which seems to be difficult to understand go on. It makes Common Lisp the lisp dialect which many persons want to be the standard. Lots of Common Lisp implementaions are developed and available now. Recently, we have a good news on the start up of the ANSI/ISO committee for Lisp standardization based on Common Lisp.This paper summerizes the process around the Common Lisp based standardization. It also describes the current issues and the author's opinion about the further movement.

1．Common Lisp の成立

Common Lisp は公式的には、CLtL[Guy84] の出版された1984年夏に始まるということができる。その胎動はその数年前からあった。Common Lisp の名が正式にあらわれたのは、[Guy82] が最初のものと思う。

Common Lisp の仕様は大きく、多数の人の意見が反映されているが、この事は、逆に、主著者であるGuy Steeleの人柄と能力を裏づけるものとなっている。また、電子メール時代において、その役割が重要かつ効果的に生かされて、言語仕様が練りあげられた最初の言語といえる。その通信メッセージ量は、CLtLの序の時点で、5.5MB(五百五十万字）という。現在、井田の手元に転送されているその後のメッセージの分量は、3MB程はある。

従来よりのLispはMacLisp 系,InterLisp系 ，そして4BSDのFranzlisp ，数式処理のPSLなどに大別できよう。これらは、いずれもCommonLispへと向かっている。Common Lisp の特徴は、豊富なそして洗練された機能にあるといってよい。それらは一朝一石に出来たのでは無く、Lispベースのソフトを蓄積してきたこれらの方言の上での経験に基づくものである。とはいえ、Common Lisp の仕様は現在まだ確定していないと見るのが正当なところだと認識している。CLtLでは定義されているはずのものが定義されておらず、各処理系作成者の意見によって作り方、設計が大きく異なるところも未だに残っている。CLtLの出版以前にKCL [Yuasa84] がほとんどCommon Lisp と呼べる処理系を発表したのは、多くの米国の処理系作成者にとって、脅威を与えたのは無理からぬことといえる。その時点では製作上のノウハウはクローズされた状態にあると多くの人が思っていたからである。

2. 世代認識によるCommon Lisp の機能の位置付けと現在の仕様の区分け

もともとCommon Lisp は、拡散する恐れのあったMacLisp の後継に対して，共通仕様を与える動機でまとめられている。しかし、今日のCommonLispへの期待には、第一に、自然発生的な共通仕様へのニーズがある。それが無ければ、今日のような動向は無かったであろう。DARPA による援助もそれを支えている。

Lispの世代を、およそ、四つの世代に分け、CommonLispの位置づけをしてみる。

第一の世代（およそ1960年代）は、マッカーシーのLisp1.5 の時代である。処理はバッチ形式で、機能も大変低い。利用者も研究者中心で、メーカのサポートも無かった。それでも、いくつかのAIソフトが作られた。

第二の世代（およそ1970年代）は、TSS の進展に伴ない、MIT やBBN （のちにXEROX)などにより開発され、その後のLispの基本となる二大潮流となるLispの時代である。今日標準的に考えられている機能のほとんどは、この時代に端を発している。例えば、多くのデータ型が導入された。配列、ファイル、文字列、ハッシュ表、N-タプル、bignumなどである。制御構造の点では、let,do,catch/throw、コルーチンなどはこの時代からあらわれている。また、プログラミング環境という視点からの具体的な道具として、構造エディタ、デバッガ、トレーサ、ブレークパッケージ、会話型ヘルプなどがあらわれた。

第三の世代は、個人用の高性能のコンピュータの上で動作するLispである。Xerox1100SIP,Symbolics3600などである。これらの上では、Lisp向きのハードウェアが可能となったために、高速にLispの持つさまざまな、実行時の処理ができるようになった。ビットマップディスプレイ、マルチウィンドウ、マウスなどが装備されている。ソフト的には具備している機能も飛躍的に増加し、オプションも増え、一つのOSと言える程の多くの機能を持つようになってきた。　これに対応して、商用の計算機の上でのLispも、重装備になり、また、移植性を良くするなど従来のものに比べ、大幅に機能の向上がなされた。

機能の向上は必然的にインタプリタの速度低下をもたらし、伝統的にLispの中心的な処理系であったインタプリタは、その位置が低くなってきた。この事は、コンパイラの役割が更に重要になることを意味し、最適化コンパイラが他の多くの高級言語のように、開発されるようになった。もちろんこの事は、Lispが高速性が要求される応用分野や、実用的な応用プログラムが多数作られるようになったこととも関連がある。
第四の世代は標準化を指向する世代、すなわちCommonLisp系のLisp処理系の出現である。この世代の処理系はまだ十分な浸透を見せてはいないが急速に人口を増している。標準化が指向されるようになったのは、1)Lispの使用者が増えた。2)メーカの製品として提供されるようになった。3)Lisp上に多数のAIソフトが構築され流通されるようになった。といったことが背景にある。

この世代認識はCommonLispの現在の言語仕様および、今後の仕様拡張を考える上で重要なものと思っている。[ida85b,d]に一部紹介したが、CommonLispの仕様の改定は十分な討議を経てのち行われることが基本的な了解となっている。この原則は、最近の電子メール討議の中でもWeinreb が述べている。

Common Lisp は第二世代までの機能はほとんど整理統合しているが、第三世代のLispの持つ機能の広がりまでは少なくとも今のところ、含められていない。たとえば、並列処理機能、マルチウィンドウ・グラフィクス、統合プログラミング環境、ネットワークインタフェィスなどである。これらのいくつかは、現在検討されている。一方、データ型や、関数の豊富さの点では、loopなどの制御構造、誤り処理、まどはまだ、不十分である。これらの諸点は委員会方式で、練りあげられてのち、仕様に組込まれる。

この結果、CommonLispの仕様は二レベルに分けられて考えられている。White pages および、Yellow pagesである。White pages が言語仕様であり、Yellow pagesはプログラムライブラリおよびプログラムコードパッケージである。Yellow Pagesには現在のところ、プログラミング環境などが入ると説明してきた。現在の時点では、環境的なソフトウェアはCommonLispの中にないということである。しかし、注意したいのは、Yellow pagesの本来の意味は、未だWhite pagesに入れるには至っていないが、かなり有用性が高くまた共通性があり、将来White pages へ組込む可能性があるもの、という点である。

なお、Common Lisp処理系の保持する文書は上記 2つに加えて、Red Pages(処理系依存部),Blue Pages（処理系ガイド）の 4つにわけて考えられる。

オブジェクト指向はWhite Pages 入りが間近だが、この他、ループconstruct ，システム生成ツール、デバッグ機能、エディタ、並列処理、論理型パラダイム、その他多くの機能が議論されている。しかし、現在主に、電子メールにより議論されているのは、膨大な仕様の正しい理解（処理系の製作という観点から）と、仕様の不備の指摘、改良・拡張提案である。筆者もいくつかの訂正案やリポートを送った。また、現在の交信録には、梅村氏、萩谷氏、湯浅氏などのメイルも登場している。

3. Common Lispのゴール　　頁数の関係で省略する。

4. Common Lispの処理系

Guy L. Steele が日本へ報告するために1985年9 月25日にリクエストし得られたCommon Lisp処理系のその時点での様子は次の通りであった。（順不同）

1) Portable Common Lisp Subset（ピクルス）1986年春　ユタ大学　有理数、複素数、文字、クロージャを除く

2)Gould CommonLisp based on spice Lisp 1986～ フルセット(Gould Power node9000 および6000 Unix 4BSD)
3)HP Development Environment for Common Lisp 1986 Jan. フルセット(HP9000- Unix systemV)
4)Tek Common Lisp 1986年第一四半期 フルセット(Tek4404-4406)By Franz.
5)ETILisp (以前のspiceLisp) 現存 perq Accent versionS6～ 複素数無し
6)Sun Common Lisp 1985年Dec～ フルセット by Lucid
7)Apollo Common Lisp 1986年Jan.～ フルセット by Lucid
8)Prime Common Lisp フルセット by Lucid
9)Symbolics Cross Compiler for Sun and Apollo 1986年March ～フルセット
10) T 現存、サブセット(互換性なし) by Yale Univ.Apollo,Vax (unix/VMS)

また、現在存在が確認されているものには次のものがる。完全に動作しているもの、開発中のもの、既に市販されているものがある。

1)DEC VAXLISP
2)DATA GENERAL Common Lisp
3)TICommon Lisp for Explorer
4)Intermetrics Common Lisp for IBM 370
5)Symbolics Common Lisp
6)ExCL Common Lisp on AT&T
7)Golden Common Lisp by Gold Hill
8)MIT NIL
9)Xerox Common Lisp
10) KCL
11)APCL

現在中心となっている処理系はSymbolics Common Lisp , VAXLISP ,KCL, ExCL CommonLisp 、Lucid CommonLisp、また、マイコン用としては、GClisp, であろう。

5.Common Lisp ミーティング

Common Lisp Meeting は筆者の知る限りでは1984年以降で4 回開催されている。
1)1984年 8月at Austin TX。出席し、[ida84] で紹介した。
2)1984年10月 at Monteley CA [ida85b]などに紹介した。
3)1985年8 月 at UCLA CA.IJCAI'85. object-oriented subcommittee 主催で開かれた。出席し、その概要は、Jeida ワークショップで述べ、要旨は[jeida86] に記した。
4)1985年12月9-11日 at Boston Marriot Hotel MA この会議は結果的に大きな意味があった。招待を受けたが出席できなかった。概要は何人かの話を総合すると次のようになる。
a)標準化について Bob Mathisを中心にANSI,ISO活動へ移行する。これについては次節で
b)Validationについて USC-ISI でtesting suite を作る。メイル討論へ
c)Error-handlingについて Kent Pitman より、Error handling機構の提案があった。メイル討論へ
d)オブジェクト指向機能について CommonLoops,new Flavors,Object Lisp,Common Objectsの説明があった。message-sending をfunction call として扱う、という重要な点で合意があった。CommonLoops のベータテスト版の開発が見守られることになった。
e)ウィンドウイング Lucid はIntellicorp と共に作っているウィンドウシステムを原案として提出する用意があることを示した。他にもいくつかの提案があった模様

6. Common Lispによる標準化

5 月7 日に受け取ったメイル(5月6 日発) では、ついにANSIにおいてX3J13 がApproveされ、本年9 月23、24日にワシントンDCのCBEMA で最初のミーティングが開かれることに

なったと伝えている。X3J13 はCommonLisp を標準とするための委員会だと了解している。昨年秋のISO のミーティングではフランスを中心として、標準Lispの制定の開始が提案された。この提案は米国預かりとなり、これを受けてANSIの下にX3J13 が作られたと聞いている。翌5 月8 日にはさらに具体的に筆者に参加を求めるメールが届いた。付録としてそれらを添付する。推進役のBob MathisはISO のADA の委員長であったらしい。ANSI/X3J13とISO/TC97/SC22 を兼ねるらしい。詳細は照会しているので後に報告する。

7.CommonLisp仕様の確定に向けて

仕様の全体像については、[ida85d],[yuasa85]など、あるいは、[Brooks85],[Win84]などのテキストを参照してほしい。また、arpaネットワーク討議の各々については、別の機会に譲る。テーマとしては本稿に含められている。ここでは、FranzLisp との対比を通して見てみる。（以下、FLとは[Franz85] のopus42を指し、4BSDに添付される旧FLと区別している。また、MLはMacLisp を指す。）

1)case sensitibity　CLおよびMLはSenseless,FLはsensitive

2)generic arithmetic functions. たとえば + is only for fixnum (ML,FLではgeneric operation is PLUS)

3)variable binding. dynamic vs lexical

4)consp 概念　FLにはlistp ならある。

5)基本条件式　　cond(ML,FL) vs　if(CL)

6)Defvar　　旧 FL には無い

7)EQL eqであれば真、あるいは同一の値の同一の型の数値であれば真、あるいは同一の文字を表す文字であれば真

```
(defmeth eql (x y) (eq x y))
(defmeth eql ((x fixnum ) (y fixnum )) (zerop (- x y)))
(defmeth eql ((x bignum) (y bignum))(zerop (- x y)))
(defmeth eql ((x character)(y character)) (char= x y))
...
```

かつ、判定を含むプリミティブの基本はこのEQL になった。(たとえば、member,assoc,subst,sublis などにおいて)

8)copy-list,copy-alist,copy-treeはML,FL には無い (copy-listはトップレベルだけをコピー、copy-alistは連想リストの形式としてコピー、copy-tree はすべてがコピーされる（従って、(subst nil nil tree)　などとしなくても済むようになった）

9) Format　ML にはライブラリとしてその一部の機能が存在　旧FLには無い

10) ファイルインタフェィス　MLのopenは必ずふたつの引数がなければならない。FLにはopenがなく、infile,outfile　というオープン関数がある

11) print　FLにはprin1 ,printがない。princ はnil を必ず返す。print が無いかわりに、必ずTを返すppがある。MLはprint,prin1,princ,terpriはT を返す。CLはterpriはnil を，他はその引数をそのまま返す

12) キーワード引数 ML,旧FLには無い

13) apply CLでは複数の引数を許す。たとえば、(apply #'max 3 5　'(2 7 3)) は7 になる。旧FLには無い

14) mapl　ML,FL のmap はmaplとなった

15) symbol-value,symbol-function,symbol-plist MLでsymbol-function 相当は、(cadr

(fboundp sym))となる。FLではgetdに等しい。symbol-plistはML,FLではplistという。新FLでは対応がとられている。
16) macroexpand-1 FLには無い
17) setf FLでは(setf (car x) y)) は(rplaca x y) と等しい
18) block,return-from,do-namedはML,FL に無い
19) 多値 旧FLには無い
20) package 機能は最新のFLには含められた。
その他FLはまだ多少変化すると思われる。MacLisp はあまり改良が進められないのではないか?

環境の閉じ込め

Common Lisp の最も大きな特徴とされていることに様々なレベルでの環境の閉じ込めがある。これは、極、普通の使い方の範囲では比較的理解しやすい。すなわち、defun された関数あるいは、function closureのbodyの中で各変数は閉じているという概念である。現在、マクロの中やmacroletでの扱い、コンパイル時とロード時の連続性などの点で閉じ込めについての議論があるように思われる。

また、更に、package の持つ機能についても議論がおきている。本当に、巨大なシステムを共同して使う時に、命名空間を切り分けるのにうまく使えるのか?zetaでのように、もっとおおがかりなサポートがいるのではないか? あるいは、package を使ってshadowing,名前の衝突を避けるという考え方の是非が議論されている。

8.電子協(Jeida) での状況

電子協(Jeida) では1985年、コモンLisp動向専門委員会の活動が行われた。CommonLisp処理系のヒヤリング、米国との対応、サブセットWG,オブジェクト指向WG、CommonLispニーズのアンケート調査などがこの一年で行なわれてきた。委員長は井田がつとめ、24の組織からのメンバーで構成されている。1985年5 月からスタートし、9 月には、ワークショップを開いた。1985年の活動の報告書は1986年3 月にまとめられ5 月中に[jeida86] として出される。アンケート結果から日本でのCommonLispの普及は企業に偏り、大学、研究機関では、KCL など以外はあまりふれられない現状などがわかる。

1986年度は名称もLisp技術専門委員会となった。サブセットWG、漢字WG、オブジェクトWG、(いずれも仮称)が各々のリーダの元に進められる予定となっている。Bob Mathisには、サブセットおよび漢字の個人的なドキュメントはリクエストがあったので送付した。今後、交流は深まるものと予想される。なおAI協会においても標準化活動が始まるように聞いている。

9.サブセット仕様について

現在日本(Jeida) および、ヨーロッパ(INRIAフランスが中心)、で議論が進められている。GLS は井田の原案に対して、コメントを書いたというメールを貰ったが、この原稿を書いている時点ではまだ届いていない。サブセットが必要な理由は[ida85e]で述べたが、大別して、1 。技術的見地、2 。教育的見地、3 。体系上の必要性、の3 つがある。

10. オブジェクト指向の組込

84年秋からの電子メール討論を経て、IJCAI'85でのObjectOrientedSubcommitteeにおい

て3 つの原案[Drescher85][Bob85][Snyder85] が出され、その中で、CommonLoops [Bob85] が有力な原案として下敷きになることになった。この時点でSymbolics はCommonLoops への賛意を表明し、対抗する原案は提出しなかった。このことは少なくとも、CommonLoops がLoops からの大きな旅立ちをしているのに対応するように、Symbolics は従来からのFlavors には固執していないことを暗に表明したものと受けとることができるものであった。その後の半年でCommonLoopsが一挙に標準として組込まれるという動きもあったが、Symbolics 系の動きのまとまりの関係から、12月の会議では、new-Flavors [Keene85] という構想がSymbolics より発表され、結論は更に、少なくとも半年伸ばされた。new-Flavors はCommonLoops と共通性が高く、メッセージ送信は関数呼出し形式をとる。

CommonLoops については[ida85f,86a][ishida86]などがあり、その詳細は[ida86b]に現在の情況をまとめているのでそちらを参照してほしい。現在、井田が直接電子的な手段で対応できる大学にはソースを送って良いというようになっているので、希望する人はjunet で、ida@u-tokyo.junet までその旨を電子的な手段で送ってほしい。

11. 今後

井田は、サブセット案を米国委員会に提出すること、仕様書第二版の作成に協力すること、CommonLoops の改良試案をXerox に提示済みなので、この面での研究を進めること、などを予定している。

ハードウェアよりの話題としては、やはり、TIのLispチップが最も注目すべきものであろう。資料によれば、CMOSテクノロジーに基づいているが、40Mhz で動作し、Lispマシンの10倍速いものを作るとされている。

LucidのPortableCommonLispは多くのマシンの上で動くが、メモリを大量に消費し、現在のところそれ程速くない。Franz のExCLはかなりの性能が見こまれている。

GoldHillがIntel と共同で作っているCubeは一応の性能を上げているようである。また、Gold Hill はそこで用いられるCommon Lisp に対して、"Concurrent Common Lisp is a trademark of Gold Hill"という主張を入れている。

言語仕様としては、本文中で述べた諸点に対する結論はこれから始まるANSI/ISO委員会で結論が予定されることになろう。これらの動きは仕様の肥大化の懸念と裏腹なので、現在日本で進められているサブセット仕様を基本水準として提案したい意図をもっており、このへんに日本が確実に貢献できそうなことがありそうに思っている。

謝辞

Csnet/junet による通信は、石田教授(東大大型計算機センタ) の助力が無ければ円滑に遂行できなかった。Arpanet メイルの受信には内外の本当に多数の研究者の助力があった。ここに謹んで感謝の意を表します。

参考文献

[Brooks85] R.Brooks : Programming in Common Lisp, Wiley and sons, 1985

[Bobrow85] D.G.Bobrow,K.Kahn,G.Kiszales,et.al. COMMONLOOPS: Merging Common Lisp and Object-Oriented Programming, Xerox Parc ISL-85-8, aug.1985

[Drescher85]G.L.Drescher, Object Lisp for experienced LISP programmers, LMI aug. 1985

[Franz85] Franz Inc. Franz Lisp Reference Manual Opus42 , Sept. 1985

[Guy82] Guy L.Steele Jr. et.al.; "An Overview of Common Lisp", Conf.Record of 1982 Lisp Conf. pp98-107
[Guy84] - ; Common Lisp the Language, Digital Press 1984
[ida84] 井田: Lispコンファレンスに出席して, bit 1984.10. no.11, pp99-101
[ida85a]- ; Common Lispは標準規格か？, 日経バイト june 1985 pp201-206
[ida85b] -; Common Lispへのいくつかの話題 WGSYM 32-5-1, IPSJ, June 1985
[ida85c]-;CommonLisp([Guy84]の邦訳 後藤英一監訳) July 1985 共立出版
[ida85d]- : 解説Common Lisp 1 ～5 , bit 1985.8月～12月
[ida85e] -; A Common Lisp Subset Proposal ,WGSYM34-4,IPSJ, Nov.1985
[ida85f] M.Ida, An Interpretation of the Common Loops specification , WGSYM35-3, IPSJ, dec.1985
[ida86a]井田、Common Lispへのオブジェクト指向機能原案の言語仕様上の特徴と問題点、WOOC'86 予稿,march 1986
[ida86b] - ; CommonLoops:Common Lisp のオブジェクト指向機能、ソフトウェア科学会、投稿中
[ishida86] 石田、井田、他、Common Lispへのオブジェクト指向機能導入の動向 WGSYM36-7 , IPSJ mar.1986
[Jeida86] 電子協コモンLisp動向専門委員会報告書、mar.1986
[Kahn85] K.Kahn, Ideas about CommonLoops& Logic programming, draft, Nov.27.1985
[Keene85] S.E.Keene and D.A.Moon, Flavors: Object-oriented Programming on Symbolics Computers, Symbolics, dec. 1985
[Kiszales86] G.Kiszales, CommonLoops meets Object Lisp, CL-Object-Oriented-Programming@SU-AI.ARPA, jan.3, 1986
[Snyder85] A.Snyder, Object-Oriented Programming for Common Lisp ,HPlabs ATC-85-1, July,1985
[yuasa85] 湯浅, 萩谷 : Common Lisp入門1 ～3 、bit 1985.4月-6月
[yuasa84],-,-; KCL(Kyoto Common Lisp), コンピュータソフトウェア、vol.1,no.2 July 1984
[Win84] P.H.Winston and B.K.P.Horn: Lisp 2nd edition, addison wesley, 1984

付録 ANSI／ISO 委員会関連の電子メールの一例

```
Sender: MATHIS@USC-ISIF.ARPA
Subject: Lisp Standardization
To: tansei!a37078%utokyo-relay.csnet@csnet-relay.csnet
Message-Id: <[USC-ISIF.ARPA] 6-May-86 10:23:43.MATHIS>
Received: from CSNet-Relay by utokyo-relay; 9 May 86 2:07:18-JST (Fri)
Status: R

This message is to confirm your net address and continuing
interest in the ANSI/X3J13 and ISO/TC97/SC22 standardization of
Lisp.  The general mailing list "Common-Lisp at SU-AI" will
continue to receive information, this special list is only for
those with an active interest in the standardization process
itself.
```

a37078 は筆者のコードである。