

~~~~~  
研究ノート  
~~~~~

ANSI X3J13 Common Lisp 標準策定合意過程 (2)

井 田 昌 之

1. はじめに

本稿は、1980 年代後半の、Common Lisp と呼ぶ、あるコンピュータ用プログラミング言語の ANSI を中心とした標準策定過程について記した研究ノートの第二部である。表記の課題に対する第一部は文献 1 にある。そこでは、米国 ANSI X3J13 での標準制定が、実質的な Common Lisp の国際標準になる過程にみられたいくつかの観察についてまとめた。本稿はこれについで、この議論の本質的な部分として Lisp1/Lisp2 の選択という問題があることを発見し、その定義とそれに付随したヨーロッパおよび日本の事情についてまとめ、最後にそれを受けた X3J13 での最終プロセスについて記述している。

2. 問題の所在: Lisp1/Lisp2

2.1 Lisp1/Lisp2 問題とは: その象徴的な課題を通して

1986 年に発表された文献 2 の論文が問題をまとめたものになっている。この議論は、プログラミング言語の設計において、記号名(シンボル)が形成する名前空間に関するものである。象徴的なテーマは次のようなものになる。

象徴的な課題: ふたつのプログラム手続き f と g があるとする。これに対して、今、仮に数学的な記法に準ずる記法を取ったとし、 f は与えた引数に 1 を加える手続き、 g は f という手続きを用いて、それに 3 を与えることで機能を果たすものとする。(手続きという用語を、関数という用語で置き換えても支障はない。) 式で書くと次のようになる。

$$f(x) = x + 1$$

$$g(f) = f(3)$$

このとき、

『 g の定義での $f(3)$ の呼び出しの f は、引数で渡したものを指すのか、あるいはその上で定義した f を指すのか?』が課題となる。

前者の仕組みを持つものを Lisp1 と呼ぶ。後者の仕組みを持つものを Lisp2 と呼ぶ。(後者の場合、 g に渡される f は実際には使われない。)

2.2 Lisp1/Lisp2 問題と呼ばれる由来

Lisp1/Lisp2 問題という名称は、それを導いた文献 2 において用いられた。Lisp1 は単一名前空間、Lisp2 は多重名前空間あるいはもっとも多いケースとしては 2 つの名前空間、を持つ、Lisp ファミリに属するプログラミング言語の分類に対して与えた名称である。

単一名前空間のシステムでは、変数の名前と関数の名前は区別されず、同一のものとして扱われ、かつ、もっとも内側で定義されている名前の値をさすものになる。したがって、 $g(f)$ の中での $f(3)$ であれば、この f は g の引数として渡された f として解釈される。

これに対して、多重名前空間をもつシステムでは、少なくとも変数の名前空間と関数名の名前空間は異なっている。 $g(f)$ の中での $f(3)$ では、 f は関数名の位置として使われていることは明らかだから、関数名の空間の中で捜される。それであれば、その上で $f(x) = x + 1$ として定義された f が発見され、その定義が適用される。

2.3 どちらが良いか?

Lisp1 と Lisp2 のどちらが良いかという選択では、次の四つの要素が基準となろう。

- A. 実行性能
- B. 表記のしやすさ、自然さ、柔軟さ
- C. 歴史的経緯
- D. 知的資産の蓄積量と利用者の人数

A. 実行性能としては、使用メモリ量、実行速度といった具体的な指標が考えられる。B. 表記の問題は、与えられたさまざまな問題をプログラムに記述する上での性能についてであり、しばしば主観的な指標が考えられてきた。C. 歴史的経緯は、文化的な枠組みとして、どのような扱いが慣習として存在してきたと判断するのか?ということである。そして、最後の問題である D. 知的資産の蓄積量と利用者の人数という要素は、しばしば政治的な活動の結果によって左右されることになる。A ないし B の点から決定できれば問題は技術的な点だけで議論できるので、自然科学的な判断基準で決定することができる。しかし、プログラミング言語の仕様の選択には、利用者の習慣とか、最初の発明者・設計者の考え方の継承、あるいは、その国・その地域での使われ方の伝統といういわば自然言語とその文化に関する話題に共通するような問題が絡んでくる。この点に非自然科学的な要素が入り、政治学的な考察が必要になってくる背景がある。

2.4 議論の時点まででの実態

最初に開発された Lisp 処理系は Lisp1.5 と呼ばれるものである。これは Lisp2 であった。したがって、その後に出現した後継は基本的に Lisp2 である。Lisp1 の代表格は Scheme という処理系である。Scheme の基本的な性質を採用して後になって開発されたのが EuLisp である。また、EuLisp はこの議論の時点では実体を持たなかった。また、Lisp1 は記述上の簡潔さを持っていたので、Lisp2 の開発者および利用者の間でも技術者の良心から、Lisp1 の設計に耳をかたむけた。Lisp2 である Common Lisp の標準化過程の中で、そこがうまくつかれたということも言える。Lisp2 の設計者は、その基本方針で産業界の標準を押し切ることをしなかった。それには他の要因もあるといえよう。たとえば、新しい設計をするのであれば、新たに国の予算が獲得できる可能性があるとか、新しいものに挑戦することを好む技術者の本性といったものも考えられる。少なくとも政治的な安定を重視した判断はされなかった。

3 Lisp1/Lisp2に関する文献の作成とその影響

3.1 X3J13 会議への提示

1986年12月1日付の X3J13 86-015 において、議長の Bob Mathis は、次の 86年12月10日から12日まででダラスで開かれる次回会議の案内を行った。その際に、重要な点として、2つのことを指摘した。1つが、ウィーンで開かれた ISO/TC97/SC22 のアドバイザリーグループの会議において、Lisp の NWI にはフランスからコンビナー、米国からプロジェクトエディターを招くことが決定されたこと、および、重要な技術的な内容についての意志決定のための論文を添付し、それらについての検討が重要となることを促した。その論文の内容は、Lisp のシステム構成法の根本的な違いに関するものであり、重要な岐路を与えるものである。これが Lisp1/Lisp2 問題である。次に、その内容がどのようなものであるのか、以下に述べる。

3.2 論文 “Issues of Separation in Function Cells and Value Cells” とその影響

表記の論文「関数セルと値セルの分離に関する論点」(文献2)は、Richard P. Gabriel と Kent M. Pitman によるものである。この著者を含めて Lisp2 の Lisp 処理系の開発で著名であり、また X3J13 の主要メンバである人たちが作成に協力していることが明記されていたので、一挙に注目されることとなった。彼らは、この論文において、論点を列記し議論を行い、Lisp2 が良いという結論を導こうとしたが、現実には、この論文が一人歩きし、さまざまな解釈があちこちで試みられ、Lisp2 である仕様の推進者が Lisp1 も公平に扱っているという点で、標準化活動に大きな影響を与えた。この内容を紹介することは、「どのような形態の分析を発表したらどういうことがおこるか」という点で貴重な例を提供しているので、その論旨と進め方を次節に述べる。

3.3 上記論文の技術的な概要

この論文の骨子は、関数セルと値セルを同一視するか分離するかと言うこと

である。どちらを取るべきかその選択肢についてこの論文では解析をしている。前半は、技術的な論点、後半は、ノンテクニカルな論点についてまとめている。

まず述べていることは、歴史的な経過である。1981年に Scheme と呼ぶ Lisp が出現し、それは、Common Lisp にも大きく影響を与えた。レキシカルスコーピングという同じ機構をもっているが、Scheme は関数と値の名前空間を統一している点に基本的な違いがある。そして、言語の中で変数名と関数名を区別しない。1986年の ACM の会議で、ヨーロッパコミュニティは EuLisp と呼ぶ設計を発表した。そして、Common Lisp も単一名前空間を採用すべきだ、という論点を論じた。Common Lisp コミュニティの多くの人は、その提案には必ずしも同意しない。けれども、その点については十分議論をする必要があるので、その調査を行なった。この論文はその結果であるとしている。

順に技術的な点に入る。まず、Scheme と EuLisp は、単一名前空間を採用した。そして、Common Lisp もそのように変更することは技術的、政治的にも望ましいことかという疑問を提示した。そもそも、Common Lisp コミュニティはそうした変更を望むのか、ということも付け加えた。つまり、Common Lisp の名前空間に関する議論は、単に技術的な選択だけではなく、Common Lisp, Scheme, EuLisp それぞれのコミュニティを段階的に統合すべきかどうかと言う大きな問題でもあることに言及した。この言及から、『技術的』と断っているにもかかわらず、論文全体の趣旨には非技術的な要素があることを読者に印象付けた。議論を進めるために、この論文では、2つの仕組みを Lisp1 と Lisp2 と呼んだ。Lisp1 は単一名前空間の Lisp であり、ここでの議論は Scheme と EuLisp がそれに該当する。Lisp2 は、関数名前空間と、値名前空間を区別するものであり、Common Lisp がこれに該当する。

次にこの論文では、歴史的な見地から Lisp2 が正統的なものだと言った。具体的には、最初の Lisp である Lisp1.5 は、1965年にジョンマッカーシーによって設計されたが、これは、2つの名前空間を持っていて、Lisp1.5の後継はすべてその性質を持っていた。すなわち、Lisp2 が伝統的には正当なものだと論じた。いくつかの技術的な論述を経て、Lisp1.5の世界では、基本的に2つの名前

空間を持つが、コンパイラの強化とともに、一部単一名前空間であることを示した。この背景に、インタープリタの動作方法に関係があることを示した。つまり、シンボルとは、関数を表し (stands for), 変数とは、値を「refer」する、という言葉の使い分けを導いた。その後の、PDP6 Lisp, MacLisp, VAX NIL, S-1 Lisp, Spice Lisp, ZetaLisp などの多数の Lisp が Lisp2 であることについて、その特徴が述べられた。そしてこれらすべてを受けて、Common Lisp が存在することを示し、その正当性を主張した。

次に、Lisp2 である Common Lisp を Lisp1 に変更しようとする際の技術的な論点についてまとめている。まず、Lisp2 では、関数の呼び出しにおいて、関数の位置と引数の位置が異なって扱われることは、エレガントではないと多くの人が感じていたことがある。この異なる扱いは、関数と値に対しては、異なる名前空間が存在することによる。その結果、評価方法が異なっており、Lisp2 は、Lisp1 に比べてより複雑になっている。Lisp2 では、関数呼び出しの関数名の位置に対しては、値名前空間の中の識別子の中の値を取り出して、関数としてそれを呼び出すことと、関数名前空間の中の識別子の中の値を取り出して、値としてそれを渡すことの2つの動作のどちらかをするようになる。値名前空間の識別子の名前を関数として使うには、Lisp2 では FUNCALL という記号を使う。ところが、Lisp1 では、それをしなくともよい。単に (FX) のようにする。また、関数名前空間の識別子の値を使うには、Lisp2 では FUNCTION という記号を使う。Lisp1 では、その必要はない。Lisp1 での表記方法は、Lisp2 のそれに比べて、簡潔になる。ここでの論点は、どちらのほうを読みやすく、処理をしやすいか、ということになる。

次に、名前の多重性、意味論としての評価、コンパイル性能および実行性能と記憶域効率、マクロの処理などさまざまな点について詳細に解析している。次に、技術的な論点として、もし Lisp2 から Lisp1 へ変更しなければならぬとした時に、どのような互換性の問題と、どのような書き換えの手間がかかるかについて論じている。いずれも自動化した方法では、100%の信頼性が得られないことを述べている。これらの議論はいかに Lisp2 を Lisp1 へ変更する

のは難しいかを示そうとしているが、すべての記述は「Lisp1 へ変換する場合どうするか」ということが前提になってしまい、読むものにとっては、たとえば、「表面上の言語の簡単さ、すなわち、Lisp1 は必ずしも処理系戦略としては簡単ではない」といった表現が途中にあっても、Lisp1 への対応をあたかも既定の路線としているかのように、容易に読み誤らせる効果をもたらした。

3.4 上記論文での関連コミュニティに関する判断

以上で、技術的な論点を終えて、次にノンテクニカルな論点として、複数のコミュニティが存在すること、および各国各地域の状況の違いの存在について、などを述べている。当時の米国でのグループがどう考えているかを表すものとして、興味深い。全体の記述は、5つのコミュニティに対する解析と、それらが統合できるかと言うこと、そして、統合にはメリットがあるか、ということの7つのセクションとしてまとめている。以下に、その要点をまとめる。

① Scheme コミュニティ

Scheme と Common Lisp の互換性はゴールではないが、Common Lisp が Lisp1 になるというのであれば、Scheme をベースレイヤ、Common Lisp をインダストリアルレイヤとした、レイヤアプローチを示唆した。Scheme コミュニティは小さなコンピュータや組み込みシステムなどを想定しているが、レイヤアプローチに興味を持つかどうかは、X3J13 が、Common Lisp を Lisp1 に変更し、例えば CommonSCHEME と呼ぶようなサブセットを決めることから始まるとした。おそらく、既存の Scheme を標準化するのではなく、新しい言語を設計するようになるだろう、と付け加えている。

② EuLisp コミュニティ

ヨーロッパで起きた EuLisp グループは、ISO に標準 Lisp として提案するような新しいスタンダードを定義しようとしている。このアプローチは、既存の仕様の上に組み立てようとするのではなく、新しい Lisp を設計しようというものである。キーメンバーは、Jerome Chailloux, Julian Padget, John Fitch, Herbert Stoyan, Giuseppe Attardi, Jeff Dalton である。EuLisp では、3 レベル

の標準を提案しようとしている。レベル 0 は、ミニマル Lisp である。第 2 のレベルであるレベル 1 は、Scheme の大きさのものであり、小さなパーソナルコンピュータにふさわしいものと意図されている。レベル 2 は、Common Lisp の大きさに対応するものであり、産業的な応用にも使えるレベルを意図している。EuLisp と Common Lisp の間のもっとも大きな違いは、EuLisp は Lisp1 であることである。Chailloux と Padget は、面談において、Common Lisp は Lisp2 であることでつまづくであろう、と言っている。また、技術的な統合が出来なければ、両者ともに歓迎していない messy な政治的な争いが生じるだろう、と述べている。

③ 日本のコミュニティ

「日本のコミュニティは、井田の報告によれば、現在の Common Lisp に同調する方向と思われる。」と記述された。商業的な市場では、現在の仕様に強いコミットメントがあり、政治的な論点は、どのような意志決定をここで行なっても、ISO のレベルでの調整をすることになるだろう、と述べている。

④ 小型コンピュータユーザーのコミュニティ

小型のコンピュータでは、Common Lisp の処理系を見つけるのは大変困難なので、Scheme を利用することになるだろう。これは、あたかも Common Lisp は商業的なニーズに合うように設計されていないかのように見せる結果をもたらしている。というのは、大きな市場である小型コンピュータの世界には、実際的な Common Lisp 処理系を見つけるのは、困難だからだ。もし、Scheme あるいはそれに類するものを、サブセットあるいは基本となる低いレベルのものとして扱うことが出来れば、そうしたコンピュータに対して、使われる Common Lisp のサブセットとなり得る。一方、小型コンピュータというものの大きさは、極めて早い速度で大きくなっている。したがって、ANSI が最終的に標準を定める時点までは、Common Lisp の仕様が「小型」コンピュータであっても、実行不可能なコストとはならないだろう、と推定している。

⑤ Common Lisp コミュニティ

Lisp ベンダーの多くは、Common Lisp と呼ぶ Lisp を使って共同して行動

しよう決心していると、コマーシャルな世界を確信させており、現在は Common Lisp を使ってコマーシャルな製品を書く時期だと確信させようとしているとした。もし、Lisp コミュニティが Common Lisp を非互換な方法で変更するのであれば、新しく確信を持ったコマーシャルな世界は、破綻するだろうし、Common Lisp の中だけで育ってきたグループは、変更があれば、強く戦うセグメントがあるだろうと警告した。ヨーロッパや日本の Lisp コミュニティを無視するべきではないし、我々の Common Lisp コミュニティを誇りを持って無視すべきではないとした。そこで、米国標準の Lisp というだけでなく、国際標準 Lisp として、そうした変更を X3 が認めることは、ベンダーがそうした変更を受け入れることをたやすくする可能性があるとした。しかし、多数のベンダーや消費者からの現在の時点でのフィードバックは、Common Lisp はすでに非公式の標準になっており、変更は意に添わないことを示唆した。

⑥ ベンダーにとって、変更は意味を持つか

多くのベンダーは、変更が数年のオーダーでコストがかかるのであれば、その変更は受け入れられないだろう。ある場合には、Common Lisp をサポートしないと言う意志決定をし、経済的にそうした変更はサポートする意味がないと思うだろう。ガブリエルは、Lucid 社では、6 人月の検討期間が必要と述べている。Symbolics Lisp にはもっと長い期間が必要だと推定されている。また Symbolics 社は、この論文に述べているような互換性の機構が、実際に提供されうようになるとは考えていない。同時に、互換性と品質と顧客に対する意向のために十分な期間が必要であり、それらには何年もかかるだろう、と述べている。

⑦ ユーザにとって、統一は利益があるか

多くのユーザは、Common Lisp でコードを多数書いている。Symbolics では、ZetaLisp から Common Lisp への転換が起きたところである。他へのさらなる変更は痛い。多くの利用者は、現在製品をビジネスとして Common Lisp 上で開発しており、彼らを変更に同意させるのは簡単なことではない。また、

X3 は、利用者を変更させるように助けたり、DARPA がそうしたユーザのためのツールに資金を出したりすることも考えられる。けれども、ある程度のユーザにとっては、例えそうであったとしても、大きな出費であり、また、非現実的であり、現在の製品出荷のスケジュールにも合わないとい認められているとした。

最後に、これらすべての技術的、非技術的な理由の両面によって、同じ方向の意志決定に到達すると結論付けている。すなわち、Lisp2 の堅持ということである。

4. 日本での Lisp1 対 Lisp2 の議論

1987年1月22日付で、JIS Lisp を検討する委員会の主査であった井田は、「JIS Lisp に関する現状認識と私見」を著し、同グループに資料として提示している。87年度に入り、井田は、電子協 Lisp 技術専門委員会の委員長に専念し、JIS Lisp 制定については、委員として入り、後に、JIS Lisp 原案制定からは退いている。この理由に関連する事項は文献1にあるが、現在必要な活動(短期的視点)がどうであっても、ISO 段階(長期視点)では、異なる立場および、異なる考慮点があったことを判断したからである。そこで次に、87年1月22日付文書を、一節をあてて、そのまま記す。原文は手書きであって、かつ、当日の会議資料として以外には参照することも入手することも困難であるので、当時の事情を記した重要な文章としてあらためて入力した。以下の文章中の EuLisp 対 Common Lisp という表現は、そのまま Lisp1 対 Lisp2 ということである。

4.1 JIS-LISP に関する現状認識と私見 ('87.1.22 井田記、原文のまま)

① 国際的状況の認識

米国での Common Lisp の標準化は、一部に技術的なつめは残されているものの、技術的な討論の段階をすませ、ANSI のもと X3J13 委員会において着々と進んでいる。現在のまま進めば、1988年初頭には標準案が ANSI でま

とまり実質的に決着すると思われる。

ヨーロッパにおける標準化は、EuLisp の仕様をまとめようとする研究段階から、各国の標準化機構、もしくは関連団体の指示を受けた公式の活動へと移行できるかどうかと言う段階にきている。ただし現在のところ EuLisp 処理系そのものを含めて、現実的な言語としての形をとるにはいたっていない。

ISO における標準化は、フランスをコンビナ、米国をプロジェクトエディタとして、その議論を開始する段階にきた。

② 上記認識に基づく私見

86 年上期において、米国は Common Lisp をそのまま ISO 化することを前提としていたように思われる。関連するいくつかの出来事があった。しかし、ヨーロッパにおける一部の熱心な研究者たちの努力が次第に姿をあらわし始め、Common Lisp に対する具体的な対案が提示されるようになった。このヨーロッパにおける EuLisp は、その実態がどうこれから形成されていくかについて、もう少し見守る必要があるが、ヨーロッパの国数を背景とした一つの位置を持ち、この結果 ISO における今後の活動に重要な影響を与えるようになった。

こうした点から一時的には EuLisp 対 Common Lisp という対決色があったが、'86 年 11 月の ISO ミーティングにおける合意等により、相互の理解、もしくは歩み寄りが感じられるようになった。

総じて、次のような共通認識があるように思われる。

『産業のニーズに対応するためにも、Common Lisp 標準の米国内での制定は価値があり、急がれるが、これをそのまま ISO 化しようとしても時間がかかり、その間 Lisp 関連の技術の進歩により、例えば 5 年後に、ISO Common Lisp が出来ても実質的な意味を失う可能性がある。それならば現在ヨーロッパで始まった動きを捉えて“Common Lisp の次”をゆっくりと検討しよう。それならば、Common Lisp の現在持っている位置・役割とヨーロッパでの活動を並行させることが出来る』

この認識は、ヨーロッパ米国共にあるように感じられる。

いずれにせよ、'86 年 11 月頃というのは、一つの時点であったように思われ

る。

③ 日本におけるいきさつ(主に標準化関連について)

日本では、Lisp の標準を求める実質的な動きを反映して、'85 年より電子協において委員会が組織された。この委員会はいくつかの事務上の判断から、マイコン技術委員会の下の特設委員会として始められた。'85 年度は「Lisp 動向専門委員会」、'86 年度は「Lisp 技術専門委員会」と呼ばれ、'87 年度には「Lisp 技術委員会」となることが予定されている。なお、'85 年度、'86 年度については、その英文名称として、「Jeida Common Lisp Committee」を用いることが、各年度の第 1 回の委員会で認められ使われてきた。この委員会は、月 1 回のペースで開かれ、またいくつかの WG を持ち、外資系を含め 30 社にのぼる組織から委員が派遣され、活発な活動をする委員会の一つであった。この委員会の大きなねらいとしては、当時より実質的に国際標準の候補となりうるものが予想された Common Lisp に対して、その標準案の作成の過程において日本からのコメントを出し、また開かれた規格とさせる点にあった。Common Lisp という名称の一般名詞化、CommonLOOPS に発するオブジェクト指向機構の標準化と日本への公開、いくつかの小さな技術的な点での反映などに実績があった。なお現在、漢字標準案(略称)の提示、産業ベースのサブセットかなどの点で寄与が期待され、またその活動が続けられている。

'86 年 7 月より JIS LISP WG が始まり、3 年間の期間が与えられた。当初は Common Lisp をベースとすることを意識する方針でいたが、①、② で述べた認識に加え、日本ではまだ必ずしも処理系がそろっていないこと、通産省からの日仏ラウンドテーブルを踏まえた、ヨーロッパを含む三極構造への意識、などにより方針の微調整を行なっているのが現段階である。

④ JIS-LISP の設計方針について

次の点を考えて Lisp 関連の育成を図るために具体的な設計の開始を行ないたい。

- 1 この JIS LISP WG で案がまとまって、それが世に出るのは早くも 89 年度、おそらくは 1990 年であろう。この時点は ANSI のタイミングよ

ANSI X3J13 Common Lisp 標準策定合意過程 (2)

- り遅く、おそらく標準的な手順での ISO 制定よりは早い。
- 2 当面の Common Lisp 対応は、電子協の別委員会で可能である。したがって、JIS LISP WG は、ISO に照準を合わせるべきである。
 - 3 ここで案を作成する「JIS LISP」が、あとにも先にもただ1つの LISP の規格と言うわけではない。また、JIS LISP が決まりつつあると言って、LISP 関連の言語仕様の研究・試作等が止まるなら、本末転倒である。
 - 4 Lisp 関連のアプリケーション・処理系開発・教育等を考えると、極力早期の標準制定が望ましい。
 - 5 米国自身、Common Lisp (現在の)が最後の Lisp 仕様だとは言っていない。数年後の改訂を想定して、ともかくも合意した優れた仕様をまとめようとしている。

以上。

4.2 日本での意志決定に関して

前述したように井田は、短期的見地から米国規格の制定に協力し、その方針を日本からも参加することで、日本の意見も取り入れたものにするのが、日本での活動、教育研究、および産業、に対して望ましいことであると判断したので、長期的な見地、そしてそれがしばしば非技術的な政治的問題の解決を必要とする、ISO レベルでの議論において、コンフリクトを生じないように JIS 策定作業からは順次降りた。

この方針を、技術的な問題を前面に出して、米国ならびにヨーロッパにアピールした。すなわち、「短期的には Lisp2、長期的な検討には Lisp1 も視野に入れる」という概念を積極的に打ち出した。そしてそれは米国、ヨーロッパの代表の意見にも反映されるようになった。この提案は合理的なものともみなされ、1986 年後半は推移し、のちに井田が離れたあとの JIS 原案作成委員会においてもこの方向を表明したのである。

以下に、いくつかのその後の JIS LISP 原案作成過程並びに、ISO/IEC JTC1/

SC22/WG16 LISP での標準化作業からいくつかのトピックを拾う。

- ① 1987年1月22日 JIS 言語標準化調査研究委員会 LISP 言語 WG 議事録より。

この会議では、以下の点が扱われた。

- a. 日仏機械情報ラウンドテーブルの窓口設置回答 (61-6-2) について、回答書の要点が報告され、それを認めた。
- b. 「JIS LISP に関する現状認識と私見」(61-6-3) (上記 3.1 節) が説明され、次の 2 点を議決した。
 1. ANSI 対応は、電子協委員会に任せ、電子協規格の制定を期待する。
 2. JIS LISP WG は、Lisp の日本案を作成し、ISO に提出する。

- ② 1987年1月6日付 X3J13 議長より井田への手紙。

この文書の内容は、1987年3月18日開催の X3J13 会議への出席を要請するものであり、JIS LISP WG1月26日議事録にも、資料番号 61-6-4 として配布され、紹介された。その内容は、日本のさまざまな Lisp 標準化と現状についての報告を求めるものであり、同時に、日本に X3J13 の資料を持ち帰り、紹介してもらうことを依頼したものである。前節の議決により、ANSI 対応は電子協で行なうことになったので、JIS LISP WG ではなく、電子協委員会にその後の資料はもたらされることになった。

5 ヨーロッパでの状況

5.1 ISO/TC97/SC22 アドバイザリ会議

表記の会議が 1986年11月11日より13日までウィーンで開催された。日本からは中田育男、徳永英二両氏が出席していると記録されている。この会議では、プログラミング言語の標準についてその進行中あるいは進行する可能性があるものに対して、アドバイスを勧告するための会議である。たとえば、PL/I, Algol60, APL, Lisp そして、POSIX などに関して勧告 (Recommendation) が出されている。Lisp に対しては、フランスがコンビナと事務局を、米国がプロジェクトエディタを出すように勧告をしている。

この時点で、フランスは実態をもった技術がその裏づけにあったとはいいがたいが、ともかくもコンビナをせよという勧告を手にいれることになる。これにより事態は混迷することがはっきりとするのである。

この根拠として使われた文献はその Document Register (ISO/TC97/SC22 N250) を見ると、N279 というもので、そのタイトルは、Proposal for a NWI for ISO/TC97/SC22 Programming Language LISP である。これ以外には、会議で配布された Lisp 関連の資料は存在しない。そこで N279 の内容は何であったのかを調べる必要がでてくる。この N279 は、文献 1 において文献 19 として紹介され、その作成経緯を 3.4 節「米国の基本方針」として参照し論述したものである。この米国の方針は、ヨーロッパ色の強い ISO の委員会において尊重はされるが、コンビナをフランスから出すということが付記されるのである。そして米国の提案は Lisp2 であるが、フランスの方針は Lisp1 であった。

次に、ヨーロッパの状況を、1) 会議の記録の点から、2) 当時の産業界の状況からそれぞれ観察する必要があるので次に述べる。

5.2 EuLisp 第 13 回会議議事録。

前述の 1 月 22 日付 JIS 委員会議事録において、61-6-5 として、湯浅氏より表記の議事録が配布された。この内容を見ると、当時のヨーロッパのグループがどう考えていたかが分かる。特に、ここでは、ウィーンの SC22 アドバイサリグループミーティングおよび、X3J13/86-010 の名前空間に関する資料について、どのようにヨーロッパが考えていたかを見ることができる。

まず、ウィーンのミーティングについて、ヨーロッパはより細かな報告をしている。ISO で取り上げる Lisp の NWI についての経過として、フランス (AFNOR) は、EuLisp に基づく提案をした。アメリカ (ANSI) は、EuLisp グループとさまざまな US コミュニティの間で、定められた言い方に基づく提案をした。また、アメリカは、Common Lisp に基づく第 2 の提案をした、と述べ、次に、SC22 は、コンビナと事務局を AFNOR が、プロジェクトエディタを ANSI がそれぞれ与える第 2 のプロポーザルを用意して、投票にか

けるよう、勧告した。そしてこれを、ドラフト DP として、1989年1月まで準備すべきだとした。標準の名前について議論があり、Lisp の標準を単に Lisp と呼ぶべきではないと考え、ISO Lisp 標準となる Lisp を、ISO-LISP と呼ぶことに異議を唱えなかった。この会議には、ANSI の主要メンバーでもある Richard Gabriel が出席しており(日本からの参加はなし)、彼は、言語仕様上の細かな点について、改良点の指摘なども行なっている。また、X3J13/86-010 に関して彼は報告をし、その文章も配布している。そして彼は、Common Lisp から関数セルを取り除くことで、Common Lisp と Scheme を統合できると、比較的分かりやすいレイヤ化ができることを強調している。同時に、関数セルを取り除く際の長所と欠点などを説明した。歴史的に見ると、2つの名前空間のほうが、より効率が良いことを指摘した。また、現実的にベンダーでの変換コストや、Common Lisp の位置について説明をし、もし、例えば5年、といった長い期間を要するのであれば、そうした変化は受け入れ可能であろう、とした。そこで、こうした長さの期間と言うのはうなずけるものだ、と言うコンセンサスを持った。ISO の活動と ANSI の活動を混ぜることを助けるために、DOD サポートを得るプランを持つべきだ、と付け加えた。また、5年と言うのは、ISO のタイムスケールにも合うことが EuLisp の議長からも述べられた。つまり、当面アメリカは Common Lisp でいき、ヨーロッパでの新規仕様の議論、そして、さまざまな活動により、5年程度で全体の統合をすると言う道筋をここで述べたものである。これは、86年後半から約半年間、井田が積極的に、米国内および日本国内で主張してきたことも符合する。なお、EuLisp の会議では、リチャードガブリエルは、オブジェクト指向拡張について、CommonLOOPS の拡張に相当する案が、87年1月中にできるだろう、と述べている。この頃、日米欧共に誰もオブジェクト指向提案採用の大きさの意味を注意するものはいなかった。このオブジェクト指向提案の採用により、Common Lisp に関する議論も、技術的な本質に関わる部分で、大きな影響を受けることになる。オブジェクト指向機能拡張は、本質的にプログラミング言語の近代化と進歩に対して大きな影響を与え、その波は、それまでに考えても

いなかったような、新しいタイプの議論の必要性を生じさせ、この結果、Common Lisp 標準化作業は当初予定を大幅に延長した期間を要することとなった。処理系製作者は、さらに数年の処理系改良期間を必要とするようになり、ソフトウェア製作者は、オブジェクト指向機能を含む形で、ソフトウェアを設計するべく、さらに長い期間がかかることとなった。Common Lisp のオブジェクト指向機能原案と並行して、C 言語へのオブジェクト指向機能拡張が提案され、それが C++ と呼ばれ、世に出るようになる。C++ は、80 年代後半から、爆発する兆しを見せた UNIX、インターネットの大きな流れの中で、そのための基本的なソフトウェアを書くためのツールとして、脚光を浴び、大きなマンパワーがその開発に費やされた。しかし、皮肉なことに、C++ のオブジェクト指向機能拡張部分は、大きく利用されることはなく、ほとんどが C 言語としてのレベルで使われていたことが後になっていろいろな資料で分かる。しかしともかくも、この頃はオブジェクト指向を採用しない限り、将来はないとさえ言われ、多くのものがそのための研究や開発の活動をしていた。Common Lisp の処理系および、応用の開発においては、まだそれ自身が安定した状態でもなく、また、小型のコンピュータ(パソコン)で十分機能する処理系がなかったのにも関わらず、オブジェクト指向と言う新しい大きな流れに取り組むことにより、Lisp を近代的な言語の中に置きつづけることはできたが、同時に、商用の十分な性能を持った処理系を広く供給することからは遠かった。この点が、Common Lisp のその後に重要な役割を与えることとなる。

5.3 ヨーロッパとの接触の中で日本の JIS 対応委員会が記したこと

井田が退いた後の JIS 原案作成委員会で進められた応答はどんなものであったか、議事録からその一部を記述する。

89 年度第 1 回会議において、委員の湯浅太一氏より、89 年 3 月 30 日および 31 日に開かれた、SC22 LISP WG の出席報告があった。その報告によれば、短期案については、日本独自のものは提案しないこと、長期案については検討すること、短期案については、日本の産業界が困らないものを、と述べたこと

を報告した。また、当時の主査より、会議全体は大変ポリティカルなものであった、と印象が述べられた。(同会議への米国からの参加者からは、日本からの出席者は発言もせず、黙っていて、何を考えているかわからない、という指摘が後に個人的にあった。)

5.4 JPAL91 でのヨーロッパ産業界の状況に関する報告

5.4.1 アウトライン

1991年11月26および27日に、社団法人日本電子工業振興協会は、ME 知的ソフト環境調査委員会の下処理系言語技術専門委員会(委員長井田昌之)の主催により、JPAL'91を開催し、ヨーロッパおよびアメリカでの現状、ならびに過去現在未来について、また、一連の活動についてのシンポジウム、および展示会を開いた。Lisp 技術の過去現在未来に関する発表者は、京都大学の長尾教授(現京大総長)である。ヨーロッパからの発表者は、ヨーロッパにおける Lisp の産業化のリーダーとして、Al Roth 氏である。アメリカからの発表者は Mark Son-Bell 氏である。ヨーロッパでの 80 年代の状況については、報告されている資料は限られているので、以下に Al Roth 氏の報告(文献3)を、原文英文から抄訳し、そのまとめを記す。

5.4.2 アル・ロス氏がとらえたその時点までのヨーロッパの状況(文献3より抄訳、ただし原文にはセクション分けは無い)

1) 全体的な認識

Lisp 言語はヨーロッパでさまざまな領域での商業的アプリケーションを記述するのに広く用いられている。Lisp が強力だと期待される分野は、ラピッドディベロップメントおよび、その展開の有効なツールとしてである。ここでは、さまざまな企業の活動が関連している。例えば、ILOG (井田注: LeLisp と呼ぶ独自の Lisp の開発者、このヨーロッパ標準ひいては国際標準化を狙うというのがフランスの根幹の戦略であった; Lisp1 系), Harlequin (井田注: 電子出版関連を主業務とした英国企業で、レイアウトから新聞印刷まで幅広い仕事を

していた。Lisp2系の Lisp Works, Common Lispなどを扱った), Procyon (井田注: 最初は独立した英国企業だったが後に Scientia に買収された。Lisp2系の Common Lisp を供給した。Lisp 部門は後に Franz 社に引き渡される), Integral Solutions (井田注: POPLOG という体系の中に Common Lisp をはじめとしたいくつかのプログラミング言語とその開発環境を供給), Delphi (井田注: イタリア企業で Common Lisp を供給)などである。Lisp に関するヨーロッパのマーケットは、アメリカに比べれば小さい(井田注: このコメントも理解に苦しむ部分がある。というのは、Siemens をはじめヨーロッパ内の多数の企業が独自の開発をしているとか、多数のアプリケーションが開発されている、あるいは Esprit プロジェクトのかなりの部分は Lisp である、等々の指摘を同時にしているからである。筆者から見たとき、これはヨーロッパ独自技術ですべてを支えておらず、米国技術が使われていることへの感情的な反発が背景にあったと感じている)。

2) 米国との関係

ヨーロッパでの Lisp は、Franz, Symbolics, Lucid などの大きな米国のベンダーによって、伝統的には支配されてきた(井田注: これらの企業は Lisp2 である処理系を販売しており、このことが Lisp1 であるヨーロッパ独自の Lisp をもりたてようとした背景にある)。このことは、多くのヨーロッパでの Lisp の活動は、そうした米国製品の上に作られてきたものであり、米国での例がヨーロッパでのマーケットをドライブする傾向があった。しかしながら、ヨーロッパの中では米国の企業は、比較すれば貧弱なサポートを与えている傾向があった。そして、過去数年間に、ヨーロッパ独自の Lisp システムのいくつかの発展をみるようになった。Lisp は多くの産業セクターで用いられている。軍事 (British Ministry of Defense がその例で、そこは大規模なユーザーである)、通信(例えば BT)、エンジニアリングと製造業などである。また、パース大学、サザンプトン大学、ケンブリッジ大学、オープンユニバーシティ、エジンバラ大学、ピサ大学、ノッティンガム大学など、多数の大学が関連をもっている。こうした広がり、Lisp のアプリケーションの多くが、実際の場に広く使われ

ていることを示している。けれどもこのことは広く認識されたものとはなっていない。

3) 標準化の認識

Lisp は 30 年前に発明されたものであるが、これはまだ十分に標準化されていない。研究コミュニティにとってはこのことは大きな問題ではないが、産業界でのデータ処理へのニーズにとってはそうではない。DARPA の支援のもとに、標準化の最初の努力が米国で、ANSI の活動とともに始まった。けれども、現在オフィシャルスタンダードを持つ Lisp の方言は、Scheme であり、それは、IEEE 標準である。標準化の努力が、なされている非標準の Lisp の中には、Common Lisp が最も大きいものである。けれどもまだ、標準ドラフトが交換されておらず、すべてのドラフトはなおも X3J13 の内部文書として存在するだけである。(井田注: このあたりの記述はいかにも Common Lisp を嫌ったものと読める)

現在現れつつある Common Lisp は、なおも普遍的な承認を得ていないいくつかの理由がある。批判をいくつかあげると以下のようなものである。システムが大きすぎる。バロクなフィーチャーを持っている。実行時ライブラリが大きく面倒である。オブジェクトシステムは、インテグレートされるのではなく、ビルトオンされている。バリデーションの手続きが十分でない。ポータビリティの問題等々等々。

85 年の 9 月に、INRIA の中において、ヨーロッパでの Lisp エキスパートの集まりがあり、EuLisp テクニカルインタレストグループ (TIG) を形成した。これは、EuLisp コミッティと呼ばれる。このグループの活動は、AFNOR, BSI, そして、DIN に発表され、1987 年 9 月以来、この委員会は CEC, コミッションオブ EC によってサポートされている。

AFNOR と ANSI は、共同して Lisp のためのワーキンググループ WG16 を 1988 年初期にフランスにおいて ISO に作った。ISO ワーキンググループの第 1 回は、そのためのコロキウムに続いて行われ、最終的に標準化のあらゆる問題について、議論した。そこでの希望は、国際的な標準となる Lisp を

作ることである。

それ以来、ISO による進行はあまりなかったが、現在の ISO のワークプランは、LeLisp、Common Lisp、EuLisp の共通のサブセットを標準化するということである。明らかにこのプランには、技術的な困難さが存在しており、特に、LeLisp と EuLisp は、根本的に Common Lisp とは遠いものである。ISO の努力は、このスタンダードがどのようなものであるべきかに関する非合意によって、停滞させられている。多くの人々は、この標準は時間のかかるプロセスとなるものと考えている。このワークプランが変更されることがあっても、驚くべきことではないだろう。これから現れる標準が、いかなる既存の Lisp の厳密のサブセットとはならない可能性もある。なおも現在のプランの元では、ヨーロッパでの Lisp が、現在の Common Lisp からの結果とは究極的に異なるものとなるだろう、と可能性がある。

4) EuLisp と LeLisp

EuLisp システムを特徴付ける機能は、クラシックな Lisp の型システムと OBJECT システムの統合であり、クラスとモジュール機構によって与えられた、抽象化ファシリティである。EuLisp は Scheme からスタティックスコーピングの性質、シングルな静的環境、そして、オペレーターおよびオペランドに対する同一的な扱いを受け継いでいる。TELOS オブジェクトシステムは、LeLisp バージョン 16 と、EuLisp の中に存在するものであり、これらのクラスの振る舞いを規定する。新しくクラスを作るプロトコル、そして、汎関数とメソッドの構造や振る舞いを規定するものでもある。TELOS は多くを CLOS から受け継いでいる。また、ObjVLisp といった他のシステムからも受け継いでいる。EuLisp では、型とクラスの間での区別は存在しない。すべてのメタクラスは、ルートのメタクラス CLASS から継承しており、すべてのクラスはルートクラス OBJECT から継承している。LeLisp バージョン 16 は、またこのアプローチを取っている。ヨーロッパでの標準活動をこれ以上述べることは、ここでの意図ではない。が、ヨーロッパでの Lisp のアプリケーション活動は、Common Lisp に限られていないことは指摘しておく。

LeLisp は INRIA で、1982 年に Jerome Chailloux によって開発された。87 年以降、LeLisp は INRIA から発した ILOG という企業によって扱われることとなった。ILOG はそれを使って、ツールや Knowledge ベースシステムを作った。その後 ILOG は、12 カ国において、3000 を越えるプロダクトライセンスを供給している。LeLisp は、750 の関数を含む、大きなプログラムの開発のシステムとして設計されている。インタープリター、コンパイラー、エディターなどのツールを持っている。LeLisp は、さらに GUI やデータベースインタフェースやエキスパートシステムの構築などの環境も持っている。LeLisp には、LLM3 と言う仮想機械言語がある。それによってさまざまなプラットフォームに移植されている。88 年に ISLisp が考慮されるようになり、LeLisp のバージョン 16 がそのために用いられるようになった。EUREKA プロジェクトの ELSY は、この LeLisp の新しいバージョンの上にプログラミング環境を作っている。91 年の半ば以降、LeLisp は TELOS プロジェクト指向プログラミングシステムを含むようになっている。顧客は、フランスにある大きな企業を含んでおり、Renault, Credit Lyonnais, Dassault, Elf, Michelin, その他である。ILOG は、国際的な成長をしており、AT&T, CSIRO, ICOT, NTT, その他でも使われている。

5) 産業アプリケーション

ここでは、当時進行中の Lisp ベースのプロジェクトについて述べる。

○エンジニアリングとデザイン: エンジニアリングとデザインに対する AI 技術の応用は広がりを見せているが、必ずしもそれらが、Lisp ベースであると用意に認識されている形をとってはいない。知識ベースのエンジニアリング (KBE) を利用することで、生産性と品質を上げ、デザインと製造工程を最適化し、デザインサイクルを短縮し、コストを削減しようとしている。現在使われている Lisp ベースの KBE ツールは、ICAD と Concept Modeler の二つである。これらは、米国で開発されたものであるが、顕著な影響をヨーロッパにも与えている。ICAD を使うことで、エンジニアリングデザインオートメーションをしている。ヨーロッパの 10 カ国で、350 システム以上使われており、

SUN, HP, Symbolics などの上で、使われている。Concept Modeler は、Wisdom Systems (Intergraph 社の傘下)からのものであり、同様にエンジニアリングオートメーションに使われている。ILOG と関連して、ルノーは、ARCHIX が開発され、車体の構造設計に役立てられている。

○データ解析: 情報機関と警察でも用いられている。BBN による知識ベースシステムである DIANA は、英国の National Drugs Intelligence Unit において、警察のデータの分析と解釈のために使われている。Watson という Harlequin 社によって開発されたシステムは、複雑な相互関連を持つ情報からなる大規模なデータベースの解析に用いられる。

○スケジューリング: Scientia による Syllabus という学校教育機関のためのリソースマネジメントシステムも存在する。教員、教室、カリキュラムなどから、時間割を作成する。これらは、フランスやドイツで用いられている。

○エアライン: 航空機業界では、AI 技術を使って、リソースマネジメントや、ゲートの割り付け、フライトスケジュールの作成、飛行ルートプランニング、搭乗員の編成、積載量管理などが行われている。ルフトハンザでは、GmbH が飛行スケジュールのために用いられている。AMS というシステムが、機体管理に用いられている。ルフトハンザでは、さらに長期のプランニングなどのシステムなどもある。これらは、Symbolics の Static OBJECT 指向データベースの上で実現されている。貨物の管理のために、スイス航空は、ARGOS というシステムを使っている。また、パッセンジャースペースコントロールでは別のシステムを 88 年 7 月から使っている。これらのシステムは、Lisp マシン上で開発され、実行されている。

○製造業: ロジカによる PFES システムは、エキスパートシステムツールであり、食品、プラスチック、薬品、その他のさまざまな企業で用いられている。製造過程の効果的な編成や、部品や所要時間の管理などに効果をあげている。英国の経済産業省 (DTI) による、製造業におけるインテリジェンスという賞で、第 2 位を獲得した。

○軍事: 軍事用には、戦闘管理システムシミュレーション、軍備のメンテナ

ンス、情勢分析、その他多数の国で Lisp ベースのシステムが使われている。Macmach という AI システムは、HAWK レーダシステムの故障診断のトレーニングに用いられている。このシステムの中核部は、シミュレーション機能である。ロックウェルは、Procyon を用いて、G-fit と呼ぶ OBJECT 指向グラフィックシステムを使い、スペースシャトルのカーゴベイに対するペイロード構成のシステムを開発している。複雑な 3次元のジグソーと言ったもののペイロードも扱えるようになってきている。手作業では何週間もかかるものを高速に構成できる。

5) Esprit を中心とする研究活動

EC コミッションは、いくつかの研究プロジェクトを動かしている。それらには、Esprit、RACE (Research Into Advanced Communication Technologies For Europe、通信ネットワークの故障診断とメンテナンスのエキスパートシステム)、DELTA (Development of European Learning Through Technological Advance、通信放送領域でのスキル学習と知識の蓄積)、BRITE (産業技術の基礎研究)、EURAM (先進型素材の基礎研究)、DRIVE (Dedicated Road Infrastructure For Vehicle Europe、道路の安全性の改善、交通の効率の改良、知的ナビゲーション)、Eureka などがある。学術研究に対する良い資金源ともなっており、同時に、産業界でのパートナーにも刺激を与えている。Esprit の中には、AI 技術が中心的に盛り込まれている。それらをすべて数え上げられないほど Lisp が多用されている。Esprit の目的は、以下の 3 点である。

1. ヨーロッパの IT 産業が 90 年代に競争力のある基礎技術を持つことに貢献する
2. IT 領域でのヨーロッパの産業界間の協力を促進すること
3. 国際的にアクセプトされる標準の開発に貢献すること

Esprit (European Strategic Programme Of Research And Development In Technology) は、約#1.5 ビリオンの予算を持つ CEC による最も大きなプログラムのひとつであり、すべての EEC 加盟国で共有されるものである。Esprit のゴールは、ヨーロッパの IT 産業がそのコンポーネントや製品やシステムの設計や

生産する能力を、高めることである。Esprit プログラムは、特定の応用領域と言うよりも、より基本的な共通的部分の R&D に焦点を当てている。(井田注: 2001 年現在 Esprit は終結している。そのウェブホームページは、www.cordis.lu/esprit/home.html にある。)

10 年計画の Esprit プログラムの第 1 段階は、1984 年に始まり、1987 年までに 3000 人の専門家と研究者が加わった。Esprit の第 2 フェーズである Esprit-II は、Esprit-I の 2 倍の規模であり、450 プロジェクト、536 組織が参加し、240 の産業界からのパートナーを含んでいる。第 3 フェーズの Esprit-III が 90 年時点で、始まったところであり、あと 4 年続く。

Esprit には、Information Processing Systems セクターは、ソフトウェア技術と高度の情報処理と言う 2 つの領域の結果として存在するものである。そこで Esprit には、次の 4 つの点に集中する。

1. マイクロエレクトロニクス、
2. 情報処理システム、
3. IT アプリケーション、
4. 基礎研究

情報処理システムセクターは、さらに 5 つに分けられる。

1. システムエンジニアリング
2. 知識エンジニアリング(知識獲得と表現の手法に焦点)
3. システムアーキテクチャ
4. ヒューマンコンピュータインタフェース
5. センサーベースシステム

6) Esprit プロジェクトの内容

Esprit プロジェクトには広範な範囲のものが含まれる。以下にいくつかを紹介する。

いくつかのプロジェクトは、ソフトウェアエンジニアリングの概念を統合して、知識ベースシステムを作ると言う部分である。KADS (知識獲得とデザインサポート)は、解析の段階でのプロセスを作るための手法論的ガイダンスをする。KADS ツールは、知識ベースのライフサイクルをカバーする手法を与えて、プロジェクト管理のサポートをするものである。Icarus は、要求仕様に対

するフォーマルな手法に関する研究である。知識エンジニアリングのワークベンチを作るためのものもある。これらは、1990年10月に始まる Vital プロジェクトに継続されていく。その他多数の Lisp および Allegro Common Lisp ベースのシステムがある。ARCHON は、Esprit-II で最大のプロジェクトのひとつであり、1400万 ECU の予算を持っている。Esprit-I の次には、実際の産業界への展開のためのプロジェクトがある。OMEGA あるいは、ESB といったプロジェクトがある。ESB は、高速プロトタイピング、モデリングとシミュレーション、故障診断、リソースプランニングなどの領域を対象としている。ESB96 は、EEC の中で知識ベースシステムのニーズがあるところでは、広く使われ、SIEMENS PLESSEY Electronic Systems で商用化されている。Esprit の成果の商用化は、始まったばかりである。

7) 結論

Lisp に関する問題は、実用的アプリケーションの開発の道具として、それほど目立っていないということである。実際に使われているが、必ずしも表立ってはいない。過去においては、コミュニティがその成功を広く知らせることに、あまり積極的でなかったことも原因である。現在では、C++ のような他の OBJECT 指向言語にも影響している。C 言語を使う大きなコミュニティが現在、OBJECT 指向化しようとしている。それは、AT&T のような強力な企業の動向とも関係がある。Lisp マーケットにとって重要なのは、短期的には、協力をすることであって、これによって競合と無駄な争いを避け、長期的なマーケットシェアの確保を目指すことであろう。けれども、アプリケーションにとっては、どの言語を使うかは、あまり問題ではない。便利なものができれば良い。ユーザの多くは、それが Lisp で実現されているかは気にしたこともないし、そのことに気がついたこともないだろう。この例は、ICAD である。良いエンドユーザは、その使い方をうまくするために、どんな言語で実現されているかを知らうとはしない。

ヨーロッパでの Lisp の将来は、どんなものになるだろうか?ヨーロッパの標準制定の将来と言う点では、EuLisp/ISLisp のコミュニティは、Common Lisp

ANSI X3J13 Common Lisp 標準策定合意過程 (2)

のユーザほど産業界との効果的なコミュニケーションがされていない、と言うことは、ほとんど役立っていない。英国国防省やブリティッシュテレコムなどは、ロンドン大学や、オープンユニバーシティなどの英国内での大学との関連の結果、Lisp の大きなユーザとなった。一方、Bath 大学、これは EuLisp の主な推進力である、は、産業界との良い関係を喜んでいない。言語としての EuLisp の状況とは無関係に、Common Lisp は、広くイギリスでは用いられつづけるであろう。

最後に、結論として、Lisp はなおもヨーロッパにおいて広く用いられている。けれども、その問題のひとつは、ビジビリティの欠如である。JPAL や LUV や EUROPAL などの会議により、Lisp ベースのアプリケーションのデリバリーを扱い、90年代の実用的アプリケーションの開発の助けとなることが重要である。

8) 上記の報告が語るもの

ヨーロッパにおいて仕掛けられた Lisp1/Lisp2 の議論は、結局、LeLisp そして EuLisp を標準のまないたに乗せるための工夫であったと考えられる。実態としては、ヨーロッパにおいても Lisp2 である Common Lisp が中心となっていたが、これに対してヨーロッパ内の新規予算を取る、あるいは米国に対する優位を作るために進められたものであろうと推理できる。が、この点については更に証言と分析が必要であろう。

6. オブジェクト指向機能拡張の採用とそれによる ANSI 最終仕様の作成

ここまで見てきたように、Lisp1/Lisp2 の選択という議論は技術的な面よりもその多くは商業的あるいは国際的な優位性に関する主導権争いにあつたといえる。これを許したのは、米国の担当者のもつひたむきな技術追求の精神と協調の精神であろう。あるいは正直すぎたとも言える。Lisp 全体の発展という点では、かならずしも十分な評価はされていないが、ともかくも、ANSI 標準の制定に向けて、すなわち、Lisp2 を選択して、日米は短期的な国際標準の作成に協力するのである。

87年2月に入り、Common Lisp Object System Specification (X3J13 87-002) が著され、それに先立つオブジェクト指向機能拡張に関する議論の収束点が示唆された。この仕様は、それに先立つオブジェクト指向機能と直接の互換性がないものであり、新しい機能が多く含まれているものである。中心となるグループは、ゼロックスパロアルト研究所にあり、実際の開発の中心者は、グレゴリー・キザレス氏、そのリーダーは、ダニエル・ボブロー氏である。これ以後、X3J13では、数百におよぶ細かな改良点に関する議論に並行し、CLOSと呼ぶオブジェクト指向機能拡張案に関して、その採用にむけて検討が進むことになった。87年度の議論の中では、オブジェクト指向を含めるかどうかという議論など、87年11月17日～18日、88年3月16日～17日の委員会にその状況を見ることができるので、次に報告する。

6.1 X3J13 87年11月17日および18日の会議

表記の会議は、コロラド州フォートコリンズにおいて開かれた。まず、87年3月のヒューレットパカードでの会議の紹介があり、次に、文字に関する問題についても説明、繰り返し機構についての説明、Lisp1/Lisp2についての説明、マクロについての説明があった。CLOSについては、その仕様書の第1章と第2章が記述されたこと、そして、第3章のメタオブジェクトに関する部分は、これから12月に作られることが述べられ、1988年2月に改訂されると報告された。処理系の検証については、ISIでの担当者 Rich・Berman が離れたので、この小委員会が活動休止になったことが述べられた。仕様の細部調整について、膨大な仕事があることが述べられた。

6.2 88年3月16日、17日の会議

この会議は、パロアルトにおいて開かれた。88年2月24日、25日のISOミーティングの報告がされた。この報告によれば、ISOでは、短期スタンダードにまず集中することが予定されていること、その出発点として、Common Lisp, Scheme, そして EuLisp になること、ISOでの標準の名前は、ISLISP

となることが述べられた。次に、CLOSの詳細についての議論が4時間をかけて議論された。その結果、CLOS仕様書1章および2章と第3章を分離することが議決された。ただし、第1章、第2章をどのようにCLTLに入れるかは含まれていない。それは今後の作業による。最終的に合意したことは、CLOSに関して、88-002 (CLOS仕様第1章および第2章)を88年4月21日までに読み、コメントを作成すること、CLOSサブコミティは、すべてのコメントについて検討し、それによって第1章、2章を改訂すること、そして、6月の委員会までに全員に配ること、6月のミーティングでは、CLOS第1章、2章を受理するかどうか議決をすること、を決めた。細部調整については、多数の議案が議決された。各小委員会からの報告がされたが、Lisp1/Lisp2については、議案とならなくなった。

6.3 88年での活動

88年では、各小委員会での細部での検討が中心となった。委員会では、毎回レターサイズで1センチの厚さにもおよぶクリーンアップの項目の議案資料が配られ、その処理に大きな時間を要した。この頃の活動には、文献1で記した文字についての議論も含まれる。また、CLOSについては、予定通り88年6月15日付で88-002Rとして、第1章および第2章の改訂版が作られた。第3章については、88年12月15日付で89-003として、CLOSメタオブジェクトプロトコル仕様書が出された。国際標準関係では、88年3月のミーティングに対して、ISO 2月24日および25日のパリミーティングの出席報告が、リチャード・ガブリエルにより提出された。日本国内では、88年6月にCommon Lispシンポジウムが開催され、その中において、現状の報告、CLOSの概要、LOOPの紹介、日本語処理のための機能拡張案、アンケートによるLisp言語の現状が述べられた。これは電子協63-A-285「Common Lispシンポジウムプロシーディング」(昭和63年6月)として発行されている。

6.4 89年での活動

89年度は、引き続き多数のクリーンアップ項目の処理がなされ、ついで、ANSI 標準文書の目次および、その具体的な記述が進められた。その元となるドラフトは、88年7月25日付で作成された。しかしそれには、CLOS 機能は含まれていない。厚さは両面印刷で8センチほどになる。これを89年度にCLOSの文書を組み込むことで改版がされ、89年8月29日版から順次文書ができていった。

6.5 DP ANSI のパブリックレビュー

1990年および91年は、あまり参加することができなかったので、詳細な報告をすることができないが、89年度の議論を基に、Draft DP ANSI が作られ、それが親委員会に回され、その後、パブリックレビュー期間に入った。パブリックレビュー対象は、X3.226-199X BSR Programming Language Common Lisp である。この X3.226-199X のパブリックレビューの期間は、当初1992年7月24日までであったが、1992年11月23日までに延長された。このパブリックレビューに対して、票決を行ない、返事を書く必要があり、そのための委員会が1993年3月3日～5日まで3日間行なわれた。この結果、パブリックレビューピリオドを終え、そこでのコメントを基に新しいドラフトを作り、セカンドパブリックレビューの期間とし、最小であれば2ヶ月間、すなわち、6月および7月に終え、全体のプロセスは93年中に終わるように意図された。93年4月より、井田は在外研究期間に入り、6月までの間は、相当の時間をパブリケーション小委員会委員長として、この最終編集作業にあてた。この期間の活動は大変だったが、後に井田の在外研究期間としてのMIT滞在がなければ、最終版ができなかったと謝辞を受けた。なお、X3J13の議長は、ここに正確な文書はないが、91、2年頃のどこかの時点で、Bob Mathis から Guy Steele に替わった。

7. ま と め

前稿と本稿から 1980 年代中期より約 10 年間の Common Lisp 標準制定に関する米国 ANSI X3J13 を中心とした日米欧のそれぞれの歩みについて、多数の資料の中から重要なポイントと思われるものを抽出し、それらについて論述した。本稿でも、資料的価値を重視し、データの整理および当時の関連する活動での動向の紹介を中心とした。この Common Lisp を中心とした標準化ならびに産業界のブームは、90 年代に入り次第に沈静化する。その理由についてはさまざまな分析をすることができようが、本稿に述べた Lisp1/Lisp2 の議論は、重大な内的な意味を持っていたと考えられる。

背景となる標準化の問題については文献 4 に述べた。産業としてのサイクルという観点も考えられる。IT はまだまだダイナミックかつ未完成な分野だといえる。この中で、各国各地域は、産業振興・技術開発競争・国際標準のリーダーシップによる優位性とグローバルな貢献，などをもとめて今もなお，本稿で述べたような活動がくりひろげられているのである。

参考文献

- 1) 井田昌之, ANSI X3J13 Common Lisp 標準策定合意過程, 青山国際政経論集 54 号, pp. 305-332, 2001.9
- 2) Richard P. Gabriel and Kent M. Pitman, "Issues of Separation in Function Cells and Value Cells" ANSI X3J13/86-010, 1986.10
- 3) Al.Roth, "Lisp in Europe", pp. 1-21, Proc. Japan Practical Applications of Lisp Forum (JPAL'91), Nov. 26-27, 1991
- 4) 井田昌之, IT 領域における国際標準策定プロセスに関する一考察, 青山国際政経論集, 51 号 pp. 357-378, 2000.9

