記　号　処　理 78－6
（１９９５．３．２４）

ISLisp (Lisp 言語の ISO 標準化案) とその動向

伊藤貴康 (東北大学)　湯淺太一 (豊橋技術科学大学)
橋本ユキ子 (日本電気)　梅村恭司 (NTT)　長坂 篤 (沖電気)　岸田克己 (NTT)

Lisp 言語の ISO 標準を設定しようという活動が，1988 年から ISO のもとに WG16 というワーキンググループを設けて行われてきている．1992 年に日本が提案した KL をベース言語とすることが決定されてから，標準化作業が急速に進展し，ISLisp とよばれる標準化案が作成され，現在 DIS (Draft International Standard) 登録のための作業が進められている．ISLisp 設計にいたる経過と ISLisp の現在の状況，ISLisp の概要について報告する．

ISLisp (a Proposal for ISO Lisp Standardization), its Background and Current Status

Takayasu Ito (Tohoku University)
Taiichi Yuasa (Toyohashi University of Technology)
Yukiko Hashimoto (NEC Corporation)
Kyoji Umemura (NTT)
Atsushi Nagasaka (Oki Electric Industry)
Katsumi Kishida (NTT)

The effort to establish an ISO standard for the Lisp language has been made by an ISO working group WG 16 since 1988. In 1992, WG 16 seleted KL, a proposal made by the Japanese SC22 LISP WG, as the base language. Since then, a remarkable progress has been made at WG 16, and a draft proposal for ISLisp was created, which is currently being processed to become a Draft International Standard. We will report the background of the ISLisp design and its current status, together with a language overview.

1 まえがき

Lisp 言語の ISO 標準を設定しようという活動が，1988 年 2 月から ISO のもとに，

ISO/IEC JTC1/SC22/WG 16 LISP

というワーキンググループを設けて行われてきている．WG16 の Convenor はフランスで，Christian Queinnec (Ecole Polytechnique) が務めており，Project Editor は米国で，現在は Kent Pitman (Harlequin Inc.) が務めている．WG16 で作業と議論に加わっている主要国は，英国，フランス，ドイツ，米国，カナダ，日本である．なお，SC22 は，ISO においてプログラミング言語の標準化を担当する委員会であり，WG16 は，その下にあるワーキンググループである．日本国内にも，対応する委員会が情報処理学会情報規格調査会のもとに，SC22 専門委員会および SC22/LISP WG 小委員会として設けられている．

WG16 の会合は，これまで 12 回開催されてきたが，1992 年 1 月にボストンで開催された第 8 回の会合において，日本が提案した KL を Lisp 言語標準化のベース言語とすることが決定されてから，Lisp 言語の標準化案を作成する作業が急速に進展した．1992 年 8 月の第 9 回および 1992 年 10 月の第 10 回の WG16 の会合と電子メールによる作業の結果，標準化の第 1 案

Working Draft on Programming Language ISLisp Draft 8.5

ができ，ISO における CD (Committee Draft) Registration のための投票が 1993 年 1 月～4 月に行われた．P-Member 国の投票結果は次のようであった．

賛成 12 （うちコメントつき 5），反対 0，棄権 1，無投票 8

この結果，WG16 としては，投票時のコメントに対応して標準化案を改訂することとなり，1994 年 6 月の米国オーランドでの会合と電子メールでの作業を経て最新のドラフト

CD 13816: Information Technology – Programming Languages, their environments and system software interfaces – Programming language ISLisp

が作成され，再度，1994 年 8 月に CD 投票が行われることとなった．同年 11 月末締切の投票結果は，次のようであった．

賛成 10 （うちコメントつき 3），反対 1 （米国），棄権 2，無投票 10

1995 年 1 月にバルセロナで開催された第 12 回の WG16 会合で，投票時のコメントを反映する作業をした後，次のステップに進むことが議論された．特に，

condition system
module system
object system 機能の拡充

について，1995 年 5 月にカナダで開催予定の会合までに作業をしてみることとなり，その結果を踏まえて，次のステップに進むこととなっている．これらについて合意の得られる良い案が提出されれば，それらを含めたドラフトを作成し，再度の CD 投票になるか，DIS (Draft International Standard) Registration の投票に進むかのいずれかになる予定である．

最終的に，ISLisp に基づく ISO 標準の Lisp 言語が成立するまでには，Lisp 言語の生みの親で Project Editor をしている米国の態度に影響される所があるが，Lisp の核部分については固まった段階に来ている．この核部分の設計は，それが日本の SC22/LISP WG の設計による KL に基づいているものであるから，日本の貢献が大きいものである．

以上のことを踏まえ，ISLisp 設計にいたる経過と ISLisp の現在の状況について以下で報告する．

2 ISLisp の設計に至る経過 — ISLisp の設計方針について

ISLisp の設計方針について説明するために，WG16 における議論の経過と日本の SC22/LISP WG における KL の設計経過を vivid に伝えることとしたい．これによって，KL とその設計方針が如何に ISLisp に取り入れられているかも説明出来ればと考えている．

ISLisp は，以下の説明からも知られるように，Common Lisp を考慮に入れて設計され，かつ言語としてコンパクトで，効率のよい処理系が作成できる Lisp 言語であると考える．

2.1 WG16 における Lisp 言語標準化の議論の経過

1988 年 2 月にパリで開催された第 1 回 WG16 会合の直前には，ISO の Lisp 言語標準案は，米国の Common Lisp か欧州の EuLisp かのどちらかを基に設計されるであろうとの予想がされていた．第 1 回 WG16 会合での議論の結果，次のような基本的合意が成立した．(WG16 議事録から原文のまま)

> The draft standard to be provided by the Working Group within 18 months will take as starting point COMMON LISP (with X3J13 cleanups) with treatments of major issues and default values to be identified (including issues in the AFNOR list and on agenda).

しかし，その後，

- 第 2 回 1988 年 7 月 Snowbird (USA)
- 第 3 回 1988 年 11 月 Brussels (Belgium)
- 第 4 回 1989 年 3 月 Fairfax (USA)
- 第 5 回 1989 年 6 月 Sophia Antipolis (France) 日本は欠席
- 第 6 回 1989 年 9 月 仙台
- 第 7 回 1990 年 1 月 Palo Alto (USA)

の会合を経ても，米国も欧州諸国も標準化のドラフト (案) が出せないという状況が続いた．特に米国は，Project Editor 国としての責任を果していないという状況が続き，担当者も 1994 年 3 月までに 3 人も交代している．(1994 年 3 月に就任した Kent Pitman は後述するように，ドイツと日本が Associate Editor となってまとめたドラフトを受けて Project Editor として積極的に活動をしつつある．) 1991 年末までのこのような状況のもとで，WG16 の存立問題の考慮もあり，1992 年 1 月にボストンで開催された第 8 回 WG16 会合で，ドイツと日本がそれぞれ，不完全ながらドラフト (案) を提出した．ドイツの案は，name space が single name space であるなど，設計方針が Scheme に依存したようなものであったため，Common Lisp を念頭において設計された日本の KL が標準化のベースドキュメントとして採用されることになった．その時点での日本案にはオブジェクト指向機能が未だ含まれていなかったので，1992 年 1 月のボストンでの決定は，次のようになっている．

1. Lisp 核言語 (Kernel Lisp)
   日本の提出した "The Kernel Lisp Language for ISO Lisp Standardization" を Lisp 核言語設計のベースドキュメントとする．

2. オブジェクト指向の核言語 (Kernel for Object Oriented Features)
   ANSI の CLOS (Common Lisp Object System) をベースドキュメントとする．

この決定に基づき (欧米諸国の強い要望で) ドイツを Associate Editor として，Lisp 言語の ISO 標準化案を作成する作業が行われることになった．第 8 回以降，次の会合や "Action" が取られている．

- 第 8 回 1992 年 1 月 Boston(USA) [base document の決定]
- 第 9 回 1992 年 6 月 Palo Alto (USA) [日本も Associate Editor に追加]
- 第 10 回 1992 年 10 月 Concordo (USA) [CD Registration 投票原案作成]

(WG16 Ad Hoc Meeting 1992 年 11 月～ 12 月： 上記原案の細部の改訂，日本は欠席)

- CD Registration の投票依頼 1993 年 1 月
- 上記投票結果の集計 1993 年 4 月
- 第 11 回 1994 年 6 月 Orlando (USA) [投票時のコメントに基づく改訂]
- CD Registration の投票依頼 (2 回目) 1994 年 8 月
- 上記投票結果の集計 1994 年 12 月
- 第 12 回 1995 年 1 月 Barcelona (Spain) [投票時のコメントの検討と今後の対応]

日本の提案である KL は，1992 年 1 月以降に改訂が行われ，その設計方針や言語としての特徴などについては，1992 年 6 月の本研究会において，その時点のものについて詳しく報告したので参照されたい [6]．第 9 回の会合で報告されたドイツの改訂案は，日本の KL を基本的な所で改訂 (例えば，multiple name space を single name space に変更) するなどの問題が多いものであったので，日本も Associate Editor とし，日本の KL 最新版をもとに見直す作業 (KL に戻す作業) をすることとなった．その作業の結果，標準化の作業が進

展し，第10回の会合で CD Registration 投票の原案が出来るまでにいたった．これ以降，ドラフトの作成と改訂の作業は Project Editor の米国を主に行われるようになっている．投票結果と第12回の会合 (バルセロナ) での要点は，“まえがき”に記した通りである．なお，Lisp 言語の標準化の活動として，米国における Scheme の標準化と ANSI Common Lisp の標準化の活動があるが，これらについては，“あとがき”で言及する．

2.2 日本の KL 設計の経過と ISLisp について

日本の SC22/LISP WG が設計してきた核言語 KL が，ISLisp のベースとなっていることを述べたが，KL の具体的な検討は，1988年2月13日 (土) 午後に品川にある NTT ソフトウエア研究所で行われた会合での議論に始まる．この会合は，2月23，24日にパリで行われた第1回 WG16 会合に先立ち，日本として長期的な視点から Lisp 言語の標準化の核言語を考える事が重要であるとの主査（伊藤）と幹事（橋本，湯淺）の判断に基づき，技術的なベースについて検討するために行われたものであった．短期的な立場からは，1984年に出版された Guy L. Steele, Jr. の本 “Common Lisp: The Language”[3] (以下，CLtL と略記) に定義されている意味での Common Lisp を，欧州の EuLisp の良い点も考慮に入れて精錬化 (refine) したものが ISO 標準になるであろうとの観測のもとに，長期的な視点から日本が Lisp 言語の標準化に貢献をすることを目標として検討することとした．

Lisp 言語とその処理系の将来を考えたとき，マルチプロセス環境，多言語環境 (論理型言語，オブジェクト指向言語，UNIX などの OS なども含め)，言語意味論の明確化といったことに対応した上で，

compact and efficient

な処理系が作成できることが必須であるとの認識に立って，核言語についての検討を CLtL を見直すという所から始めた [1]．CLtL でも大き過ぎる，米国の X3J13 という ANSI Common Lisp 標準化作業グループで目指しているものはさらに巨大化しているというのが基本的な認識で，CLtL から削り取れるものは可能な限り削り取るという作業を CLtL の企業での使用経験と CLtL の処理系作成の経験を基に行った．KL の言語としての規模とイメージを，基本的には，この会合で与えている．

1988年4月からは，SC22/LISP WG のもとに，核言語設計の Ad Hoc Group を設け，詳細検討と文書化の作業を始めた．1991年まで，このグループの議論に加わった主要なメンバーは

湯淺 (主査)，伊藤，橋本，黒川利明（日本ＩＢＭ），梅村

であった．このグループの作業結果が，ドキュメントとして初めて公表されたのは，1989年9月に仙台で行われた第6回 WG16 会合においてであった．仙台ドキュメントを改訂したものが，1992年1月にボストンで行われた第8回 WG16 会合に提出され，ISO における Lisp 言語標準化のベースドキュメントとして採用された

The Kernel Lisp Language for ISO Lisp Standardization

である．このドキュメントと同じ会合に提出されたドイツの提案を比較すると，日本の KL は CLtL を基に設計されたコンパクトな言語であるという点で評価され，技術的な観点からベースドキュメントとして採用されたが，標準化に不案内な大学関係者と研究者が文書化を行ったために，用語の定義や体裁上の問題があった．また，CLtL から削り過ぎになっているとの意見が出され，その後，諸種の改良と拡張が日本と WG16 において行われ現在にいたっている．WG16 における改訂作業の概要については，2.1節に記した通りである．

日本独自の改訂拡張について言及しておきたい．1992年1月の第8回会合の後，SC22/LISP WG としても改訂拡張作業を行い，ドイツに連絡したが，name space が multiple であったのが single に変更されるなどの問題が発生した．これらの多くは，WG16 のその後の作業で改良改訂され，KL に近い形に改められている．また，module system と object system についても長期的な観点から (湯淺，伊藤で) 予備検討をしていたこともあり，1992年8月に，これらを含めた KL のこれまでの所での最終ドキュメントが作成され，WG16 の資料として登録している．この KL 最終版には，(不完全ながら) module system と object system も核言語の KL にマッチするようなコンパクトな提案がされている．なお，第12回 WG16 会合 (バルセロナ) で課題となった module system は，この時の module system をもとに日本 (担当：湯淺) が原案を作成することとなっている．しかし，これまでの実用的な Lisp 言語で module system を備えたものはなく，したがって，モジュール機能の使用経験も Lisp にはないので当面の ISO 標準に module system が加えられることはないであろう．

3 ISLispの概要

本節では，昨年11月の第2回CD投票に付された最新のドラフトに基づいて，ISLispの言語概要を説明する．付録に最新ドラフトの目次を掲載するので参考にされたい．なお，最新ドラフトの全文は

ホスト `katura.tutics.tut.ac.jp` の `/pub/islisp.ps`

からanonymous ftpできる．ちなみに，KLの最新版も同じホストの `/pub/kl.ps` からanonymous ftpできる．

3.1 ISLisp設計のゴール

ISLispを具体的に設計することになった第8回WG16会合の時点での設計のゴールは，KLをLisp言語の核言語設計のベースとし，Common Lispやドイツの提案(以下，DKLと呼ぶ)の機能を追加してコンパクトな核言語を設計し，オブジェクト指向機能は，CLOSをベースドキュメントとして，核言語にフィットしたコンパクトなオブジェクト指向機能を設計することであった．その意味では，現在のISLispは，この目標をほぼ達成したものになっている．最新のドラフトでは，そのIntroductionで設計のゴールについて次のように書かれている（原文のまま）．

> The programming language ISLisp is a member of the Lisp family. It is the result of standardization efforts by the committee ISO/IEC JTC 1/SC 22/WG 16.
>
> The following factors influenced the establishment of design goals for ISLisp:
>
> 1. A desire of the international Lisp community to standardize on those features of Lisp upon which there is widespread agreement.
> 2. The existence of the incompatible dialects Common-Lisp, EuLisp, Le-Lisp, and Scheme (mentioned in alphabetical order).
> 3. A desire to affirm Lisp as an industrial language.
>
> This led to the following design goals for ISLisp:
>
> 1. ISLisp shall be compatible with existing Lisp dialects where feasible.
> 2. ISLisp shall have as a primary goal to provide basic functionality.
> 3. ISLisp shall be object-oriented.
> 4. ISLisp shall be designed with extensibility in mind.
> 5. ISLisp shall give priority to industrial needs over academic needs.
> 6. ISLisp shall promote efficient implementations and applications.

3.2 Object System と Class

ISLispのobject systemは，概念的にはCLOSと同じで，CLOSの機能を大巾に縮小したものと考えてよいであろう．object system関係で定義されている関数類は次のものだけである．

`call-next-method`	`defclass`	`initialize-instance`
`class`	`defgeneric`	`instancep`
`class-of`	`defmethod`	`next-method-p`
`create-instance`	`generic-function-p`	`subclassp`

metaclassの概念は存在するが，metaclassを定義する機能は提供されていない．多重継承を許しているが制約がある．あるclassを定義する際に2つ以上のdirect superclassを指定できるが，2つのdirect superclassが，同じclass（後述の `<standard-object>` と `<object>` を除く）をsuperclassとしてはならない．この制約によって処理系は簡単になり，仕様記述も簡潔になるが，不自然な制約であり，処理系による静的なチェックも困難なために，現在検討中（制約を排除するか，それとも多重継承そのものを禁止するか）である．

図1にpredefined classとそれらの継承関係を示す．`<object>` はすべてのobjectが属するclassであり，`<standard-object>` は `defclass` で定義されたclassのinstanceが属するclassである．`<standard-class>`

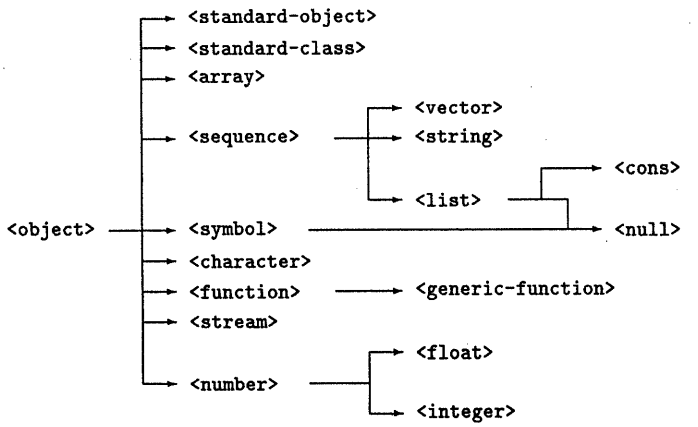

図 1: Inheritance Graph

は defclass で定義されるすべての class が属する metaclass であり，図から，class は object であることがわかる．Common Lisp と同様に，ISLisp でも，空リスト，記号の nil，真偽値の「偽」は同じ object である．predefined class の <null> はこの object だけからなる class である．

図では，<array> と <vector>，<vector> と <string> の関係が記されていない．現時点の仕様では，これらは処理系依存 (implementation defined) となっている．しかし，このままではプログラムの移植の際に問題が生じる．日本と米国は，<array> を <vector> の superclass とし，<vector> を <string> の superclass とすることを主張したが，多重継承そのものに反対するフランスと意見が対立し，いわば妥協の産物として現状のようになった．なお，フランスは <null> が <list> と <symbol> の両方の subclass であることにも反対を表明している．

3.3 Form と評価

ISLisp は multiple name space を採用しており，変数，動的変数 (dynamic variable)，関数，class，block，tag の 6 つの name space を持っている．動的変数とは，dynamic binding の対象となる変数であり，indefinite scope を持ち，dynamic extent を持つ．その他の変数 (以下では静的変数とよぶ) は lexical scope を持ち，indefinite extent を持つ．Common Lisp では，動的変数と静的変数は name space を共有していたが，これらの name space を区別することによって，ISLisp はプログラムの visibility を高めている．

form は，リスト形式で与えられる compound form，記号で与えられる identifier，その他の literal に分類される．compound form はさらに，defining form，special form，macro form，function form に分類される．defining form とは，次の名前で始まる form で，トップレベルのみで許される．

```
defclass     defdynamic  defglobal  defmethod
defconstant  defgeneric  defmacro   defun
```

defdynamic は動的変数を宣言するもので，Common Lisp の defvar に相当する．defglobal は静的変数を定義する．

special form には次のものがある．

```
assure       flet      let          tagbody
catch        function  progn        the
class        go        quote        throw
convert      if        return-from  unwind-protect
dynamic      labels    setq         while
dynamic-let  lambda    setf
```

Common Lisp の special form あるいは組み込みの macro form と同じ名前のものは，Common Lisp と同じ機能か縮小された機能と思ってさしつかえない．while は繰り返しのための form で，C 言語の while 文と同様に，条件式と本体からなる．(lambda ···) は Common Lisp の (function (lambda ···)) に相当する．the と assure は，いずれも Common Lisp の the に対応するが，与えられた式の値が指定された class に属さない場合に，assert はエラーを起こし，ISLisp の the の場合は動作が保証されない．つまり，assert はエラーチェックのために使用し，the は宣言的に使用することができる．convert は型変換を行ない，Common Lisp の coerce に相当する．

dynamic-let は動的変数の binding を行なうもので，dynamic は動的変数を参照するためのものである．動的変数への代入は (setf (dynamic 変数名) 値) の形で行なう．静的変数の binding は let が行ない，form として identifier が与えられれば，静的変数への参照となる．静的変数への代入は (setq 変数名 値) とする．

マクロに関しては，defmacro という defining form が定義されているが，その詳細，特にマクロ展開の際の環境については (マクロに関する第 8 章がわずか 2 ページしかないことから想像できると思うが) まだ決まっていない．組み込みのマクロには次のものがある．

```
and          cond  with-error-output
case         for   with-standard-input
case-using   or    with-standard-output
```

case は比較に eql を使い，case-using は比較に使う述語を指定できる．for は Common Lisp の do に相当するが，do が implicit tagbody と implicit block を設定するのに対して，for はこのどちらも設定しない．do の設定する implicit tagbody と implicit block は，実際にはほとんど利用されないので，インタープリタのオーバーヘッドを非常に大きくするだけ，と批判されてきた．

with-error-output，with-standard-input，with-standard-output は，標準の error stream，input stream，output stream を，指定される stream に切替えて本体を実行するものである．Common Lisp では，例えば *standard-input* という special 変数が組み込みで用意されており，この変数を binding しなおすことによって，同様の機能が実現されていた．ISLisp の場合は，組み込みの変数は一つもない．もちろん *standard-input* に相当するものもない．だから，専用のマクロを用意するのである．

組み込みの定数には次のものがある．

```
*most-positive-float*  nil   t
*most-negative-float*  *pi*
```

名前から，意味は容易に想像できるであろう．

4 あとがき

Lisp 言語の標準化案である ISLisp の概要とその設計に至る経過および現状について説明してきた．Lisp 言語の標準化の活動としては，この他に米国における Scheme の標準化と ANSI Common Lisp の標準化の活動がある．Scheme の標準化は 1984 年に始まり，いくつかの仕様案を経て，1988 年に revised[4] report と呼ばれる仕様書 [4] が作成された．これを受けて，IEEE の MMSS (Microprocessor and Microcomputer Standards Subcommittee) で標準化作業が行なわれ，1990 年に承認されたドラフト [5] が，1991 年に IEEE 標準となった．一方，Common Lisp の標準化は CLtL 出版直後の 1985 年にすでに始まり，ANSI の X3 委員会のもとに X3J13 という小委員会で審議が続けられ，昨年ようやく標準仕様が認められた．現在，最終仕様に向けて些細な editing 作業を行なっており，近く正式に ANSI 標準仕様が発表される予定である．当初の予定では 1988 年に標準化作業が完了する予定であったが，巨大な言語であるために，標準化作業に手間取ったのである．

ISLisp については，1995 年 1 月末にバルセロナで開催された最も最近の第 13 回 WG16 会合では，CD Registration 投票時におけるコメントについての技術的検討に加えて，今後の方針に関する次のような事柄が検討された．

1. ISLisp の改訂と機能の追加 (condition system など) の検討と作業を早く済ませ，次のステップである DIS 投票に進むこと．

2. 簡単に処理の済まない事項 (module system，object system 機能の拡充，international character set handling[2] など) は，ISLisp の ISO 標準化の後，ATTACHMENTS として処理すること．

3. 米国の ANSI Common Lisp を ISO における Fast-Track Procedure によって処理をし，ISO Common Lisp にすること．

これらについての検討と今後の具体的な方針の決定は，1995 年 5 月末にオタワ市 (カナダ) で開催予定の第 13 回 WG16 会合で行われる予定であるが，今後は，

> 先ず，ISLisp の標準化の作業と投票などの手続きを済ませ，その後，合意が得られるようなら ISO Common Lisp の Fast-Track Procedure を取る

という方向に進むことと予想される．

ISLisp の最終決着までには，さらに，技術的な詰めも含めて 1 年余の日数が必要とされると思われる．しかし，これまで説明したごとく ISLisp の核部分は出来上がった状態にある．この設計には KL を設計してきた日本が重要な役割を果して来たのは，これまで説明した通りであるが，これまで ISO で SC22 が担当してきたプログラミング言語の標準化に日本の案が標準化案やベースドキュメントとして採択された例はないとのことである．ISLisp に関する日本としての貢献を記号処理研究会の最終回に説明するのも意義のあることと考え本発表をさせて頂く事とした次第である．

最後に，SC22/LISP WG の主査 (伊藤) から Lisp 標準化について一言．Lisp 言語は，1950 年代末に提案され，FORTRAN に劣らない長い歴史を持つ言語である．Lisp は専ら研究者の間でのみ使用されてきた言語であったが，1980 年代になって CLtL が出版されたのと，丁度，バブル全盛期における Lisp を用いたエキスパートシステムによるビジネスチャンス開拓のニーズから，Lisp を industrial language として標準化したいという要求が高まった．その結果，1984 年～ 1987 年にかけて国内外で標準化のワーキンググループが始まることになった．しかし，Lisp 言語やその処理系の研究者，Lisp を用いて人工知能システムの作成をしている研究者にとっては，

> Lisp is a language that should not be standardized.

という考え方を持っている人が多数いるのではないかと考える．これは，Lisp が依然として，研究者のための言語であり，Lisp が言語として evolutionary な性質を持った言語であるということに起因していると考える．Lisp を熱心にやってきた若手の研究者や技術者に引張り出されて付き合うことになったが，SC22/LISP WG の主査もこれに近い考え方を持っている．Lisp の将来を考えると，ISLisp，ISO Scheme，ISO Common Lisp，...というように，Lisp 言語のファミリーの標準化が行われていくことにより，特定言語の国際標準化による押し付けという弊害が生じることがないことを期待している．

日本における Lisp 言語標準化の活動は，1986 年に電子協における調査委員会以来多数の方の協力と援助を得て進められて来た．1994 年時点での情報処理学会情報規格調査会のもとでの SC22/LISP WG のメンバー構成は次の通り（名簿順）である．

主査：伊藤貴康 (東北大)
幹事：橋本ユキ子 (日本電気)，湯淺太一 (豊橋技科大)
委員：浅井登（富士通），梅村恭司 (NTT)，岸田克己 (NTT)，黒川利明（日本ＩＢＭ），
田村実（日立），長坂篤 (沖電気)，元吉文男（電総研）

1995 年 1 月には，山形蔵王で会合を持ち，この発表を行うことを決めたが，その時の出席者が，本資料の連名者となっている．伊藤，湯淺，橋本，梅村は，このワーキンググループ発足時からの standing member と言えるメンバーで KL の設計と文書化にも携わって来た．長坂，岸田は，最近数年間，ISLisp ドキュメントの検討作業に加わっている．なお，本資料の第 1，2，4 節は主として伊藤が執筆し，第 3 節は主として湯淺が執筆した．

謝辞

Lisp 標準化活動を支援していただいている情報処理学会規格調査会 SC22 専門委員会の前委員長の中田育男先生と現委員長の土居範久先生，情報処理学会規格調査会の斉藤彰夫氏，及川清氏，加藤良子氏，峰崎幸子氏に感謝の意を表します．

参考文献

[1] Takayasu Ito and Taiichi Yuasa, Some Non-standard Issues on Lisp Standardization, Proceedings of the First International Workshop on LISP Evolution and Standardization in Paris, IOS, 1988.

[2] 黒川利明，湯淺太一，橋本ユキ子，梅村恭司，伊藤貴康，LISP における国際文字処理方式に関する技術的諸問題，情報処理学会研究報告 89-SYM-51，1989.

[3] Guy L. Steele Jr., Common Lisp the Language, Digital Press, 1984.

[4] Jonathan Rees and William Clinger (eds.), The revised^4 report on the algorithmic language Scheme, 1988 ACM Conference on Lisp and Functional Programming で配布，1988.

[5] IEEE Standard for the Scheme Programming Language, IEEE Std 1178-1990, 1990.

[6] 湯淺太一，梅村恭司，橋本ユキ子，黒川利明，伊藤貴康，Lisp 国際標準化のためのベース言語 KL について，情報処理学会研究報告 92-SYM-65，1992.

付録：ISLisp ドラフトの目次