コンピュータと教育　24－1
（1992. 11. 20）

Common Lispインストラクションシステム

太田信康　江藤 香　丹羽次郎　松田郁夫
日本工業大学

最近，プログラミングについて，エキスパートでなくてもプログラマがプログラミングを行う必要な場合がある．本研究では，プログラミング経験が少ない初級者に対して，的確，簡潔な情報の提供することを目標として，プログラミング中にユーザを支援するための知的情報処理システムの構築を行った．支援内容として，言語の支援にとどまらず，プログラム開発の上流過程すなわち，プログラムの前段階である機能分析の支援について考察した．ユーザがプログラミングする問題より，幾つかの単純な命題を抽出し，その命題よりシステムが用意した処理とのマッチングを取ることで，機能の明確化およびプログラミング技術の支援をおこなう．同時に，言語の支援をおこなうことにより，プログラミング中にユーザを支援するシステムのプロトタイプを開発した．

Common Lisp Instruction System

Nobuyasu Ohta　Kaoru Eto　Jiro Niwa　Ikuo Matsuda

Nippon Institute of Technology

In this study, we develop intelligent instruction system that supports its users to present fitting or brief information for beginner who is short of programming experience, during they are programming. For contents of instruction, we think not only programming language but also a function analysis. We get some simple procedure from this problem has to program. This system instruct programming technique by matching simple procedure with the procedure of this system. We were able to develop such a prototype of instruction system.

1．はじめに

近年，コンピュータに関する技術革新により，低価格高機能が現実化され，さまざまな分野にコンピュータ化が浸透して，今までに一度もコンピュータを触ったことがない初心者でも，コンピュータを利用する機会が増えている．またプログラミングについてもその言語のエキスパートでなくても，プログラマがその言語を使用しなくてはならない場合がある．そのようなとき，プログラミングの支援をおこなうシステムが必要である．

本研究の目的は，プログラミングの経験が少ない初級者に的確，簡潔な情報を提供することを目標として，プログラミング中にユーザを支援するための知的情報処理システムの構築である．従来からプログラム作成を支援するＣＡＩの構築についての研究は行われていたが[1-2]，これらの研究ではプログラム言語についての支援であり，プログラム開発の下流過程での支援に限られていた．本システムでは初級者にプログラム作成の支援を行うにはプログラム言語の支援だけでは不充分であり，プログラム開発の上流過程での支援が必要であると考えて，プログラム言語の支援だけでなく，その上流過程の支援を試みた．

本システムの具体的な対象は，人口知能の研究において盛んに使用されているＣｏｍｍｏｎ　Ｌｉｓｐ言語を取り上げた．ユーザはＬｉｓｐ言語について参考書は読んだことはあるがプログラミングの経験は少ない初級者に限定した．

本論文では，第２章ではシステムの概要，第３章は機能分析，第４章はプログラム作成の支援，第５章は支援例について，それぞれ述べる．
本システムの支援範囲，支援内容，支援方法，知識の提示などについて報告する．

2．システムの概要

初級者がプログラム言語などの参考書を読んだだけで，プログラムの作成するのは不可能に近い．身近に質問に対して答えてくれるエキスパートがいれば，それが最良の支援環境であろう．しかし，常にエキスパートがいてくれるとは限らない．そこで，コンピュータがエキスパートの代わりをすることが望ましい．

2-1　支援範囲

コンピュータによる問題処理の手順として，図１のような手順が考えられる．図１の実行，デバッグについては最近のソフトウエアにオンラインヘルプのような，エラーの説明，命令の説明をユーザに提示する機能が多く見られるので，ここでは支援対象としない．したがって，問題処理の手順で支援すべきは次の３つが考えられる．

(1)要求分析の支援
(2)機能分析の支援
(3)プログラムの作成の支援

(1)要求分析の支援は言語には直接関係なく問題の明確化，全体の計画などをおこなうことである．この問題の対象領域は膨大であるので，今回は取り扱わないことにする．

本システムでは，(2)機能分析の支援，(3)プログラム作成の支援について取り扱うことにする．

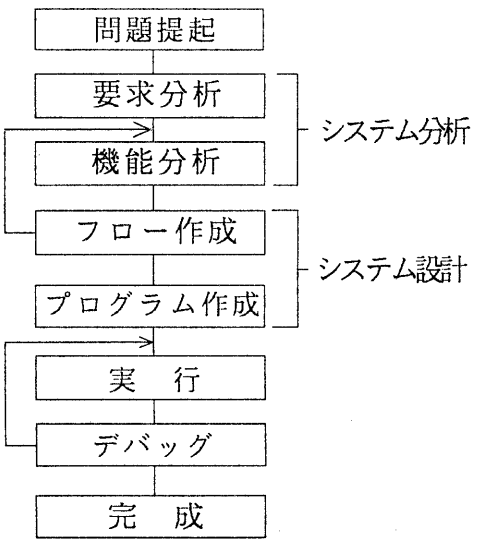

図1　問題処理の手順

2-2　システムの概要

システムの構成を図２に示す．本システムでは，前述の支援範囲より(2)機能分析の支援，(3)プログラム作成の支援に限定したので，それらの支援についての操作とシステムの流れを別々に説明する．

2-2-1 プログラム作成の支援

①ユーザが自然言語（ヘボン式ローマ字）によりシステムに入力をおこなう．

②システムが関数を推論する際，ルールデータベースより検索をおこなう．

③検索結果を要求データとしてシステムに貯え，推論を繰り返す．

④推論結果に基づき知識データ　ベースより検索をおこなう．

⑤ユーザに知識を提示する．

⑥システムが解釈不能な要求な　どに関する応答をおこなう．

2-2-2 機能分析の支援

⑦マウスでルール（単一の処理）　群より選択をおこなう．

⑧システムがプログラミング技術を推論する際，ルールデータベースより検索をおこなう．

⑨検索結果を再びルール群として，処理データをシステムに送る．

⑩ユーザに再びルール群の選択をおこなってもらい，推論を繰り返す．

⑪知識の検索，提示については，前述の④，⑤に従う．

前回までの研究では[3]，(3)プログラム作成の支援について検討した．その結果，関数の使用方法及び基礎的なプログラミング技術の支援をおこなえた．今回の研究では，さらに(2)機能分析の支援の部分，つまりプログラム作成の前段階について支援を追加する．

これらの支援をおこなうためには以下の点が重要である．

①ユーザの要求に的確な知識の提示

②ユーザが効率良く情報獲得をおこなえる

③プログラミング中に支援ができる

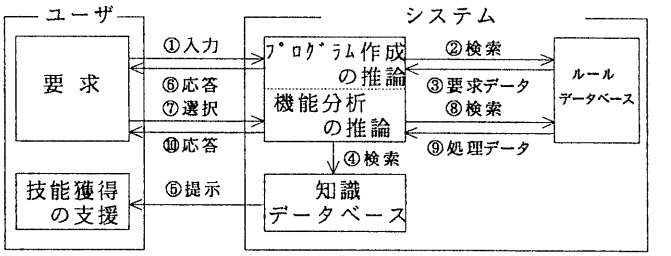

図２　システムの概要図

これらを実現させるにはエディタ上でプログラミング中に知識の提示をする必要がある．また，ユーザの要求をシステムに入力する方法も幾つかあり，要求に的確な知識を推論するためには，最適な入力方法を考える必要がある．以上の事からインターフェイス及び推論は重要な要素であると考えられる．

３．機能分析の支援

3-1 機能分析

機能分析とは問題を処理するための機能の明確化をおこなう項目である．機能分析の支援は問題処理の手順より，要求分析から問題は明確化されていると考えプログラミングに入る前段階すなわち，プログラミングのアプローチの部分について考える．

支援範囲として，関数の使用方法，プログラム作成の支援については，第４章のプログラム作成の支援の部分で支援をおこなうので，機能とその

順序を決定する具体的な手順について支援をおこなう．具体的な手順というのは，単一の処理と全体的な流れのことである．単一の処理は，単純なプログラムのことで幾つかのプログラミング技術によって構築される．具体的な手順の全体的な流れについては，単一の処理の組み合わせなので，単一の処理がわかれば，ユーザが全体的な流れを構築できる．これより，全体的な流れについてはユーザに構築してもらい，単一の処理について支援を考える．機能分析の支援は，単一の処理すなわちプログラミング技術の支援をすることにより，ある程度，支援できると考えられる．ここで，具体的な例を使用して支援方法について述べる．

3-2 支援方法

問題として「住所録を作り，任意の人物のデータをいつでも表示できるプログラムを作成する」があるとする．この問題を解くには，”入力”，”出力”，”データベース”，”検索”，”文字列操作”などの幾つかのプログラミング技術が必要である．ここで本システムの対象者はプログラミングの経験がほとんどないようなユーザなので，問題よりこれらのプログラミング技術を利用してプログラム作成をするということがわからない場合がある．しかし，「住所を登録する」，「任意の人物の住所を取り出す」，「住所録を表示する」など必要な事項は，要求分析の項目で明確化している．機能分析の支援として，この必要事項より機能の明確化及びそれに伴うプログラミング技術の説明の提示をおこなう．

3-2-1 インターフェイス

インターフェイスには大きく分けて２種類考えれる．

ａ）自然言語により要求を入力してもらう方法

ｂ）システムより選択肢を与え，ユーザに選択してもらう方法

ａ）自然言語による入力

上記の問題で，ユーザは「任意の人物の住所を取り出す」といった処理を考えた場合，このまま自然言語で入力しても，この要求は特定の問題についての要求なのでこの要求に対する機能およびプログラミング技術を提示するのは難しい．ユーザはプログラミングの経験がほとんどないようなユーザなので，システムにあった要求を入力するのは困難である．

ｂ）選択による方法

上記の問題で「任意の人物の住所を取り出す」といった処理をおこないたいと考えた場合，シス

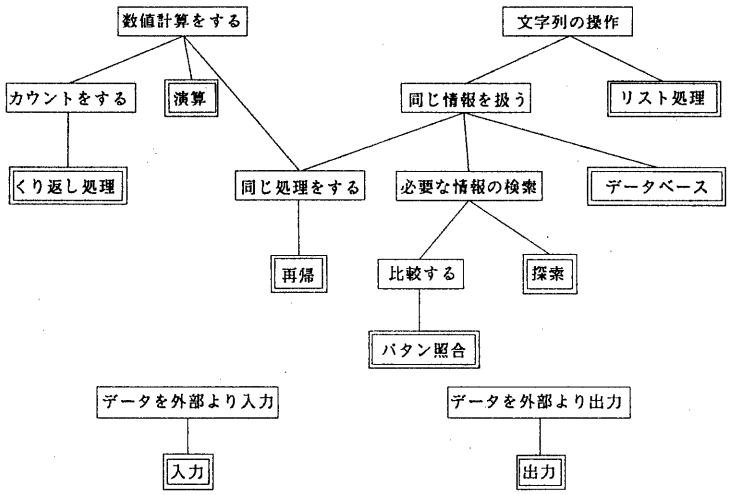

図３　プログラミング技術の構造図

テムから「必要な情報の検索」などの処理を選択肢として表示があれば，ユーザが必要な処理を選ぶことにより，ユーザの要求に対する知識の提示がおこなえる．ここで，本研究においてユーザは参考文献4-5)について読んだことがある者なので，前述の選択肢がどのような処理をするかは理解していると考える．したがって機能分析の支援のインターフェイスとしては，選択の方法を採用する．さらに，実際的には選択方法としてマウスによる選択を考える．

3-2-2 選択するルール（選択肢）

システムが単純にプログラミング技術の項目をルールとして提示するだけでは，ユーザには自分の問題処理に適合する項目がどれであるかわからないと思われる．そこで，前述の「必要な情報の検索」などの単一の処理をルールとして設ける．本研究では参考文献4-5)を参考にＬｉｓｐ言語の特徴を活かして文字列操作，関数定義，再帰などを中心にルールを考えた．図３の構造図は単一の処理とプログラミング技術の関係を表した図である．この構造図の一重枠をルールとして，システムがウインドウに提示し，ユーザはマウスにより選択することで，ユーザの要求に対するプログラミング技術の説明を提示する．

3-2-3 推論の方法

図３の構造図を利用して，推論する方法を具体的な例を持って説明する．上記の問題で「任意の人物の住所を取り出す」といった問題があり，ユーザがこの問題を解こうとした場合，まず，図３のトップレベルのルールとして，「数値計算をする」，「文字列の操作」，「データを外部より入力」，「データを外部に出力」の項目をウインドウに提示し，マウスで選択をおこなう．この問題では「文字列の操作」なので，このルールを選択する．選択すると階層の「情報を扱う」，「リスト処理」という２項目に分かれる．さらに選択をおこない「情報を扱う」を選択，さらに「同じ処理をおこなう」，「必要な情報の検索」，「データベース」に分かれ，「必要な情報の検索」を選択，最後に「探索」という項目に進み推論を終了させる．このようにしてユーザの要求に対するプログラミング技術を推論し，知識の提示をおこなう．また，知識の提示後，関連のある関数やプログラミング技術がある場合，さらに説明が必要ならば，選択して説明の提示をおこなう．

3-3 支援内容

図３の二重枠のプログラミング技術について説明をおこなう．さらに，具体的な事例，関連のある関数について説明の提示もおこなう．このときの関数の説明は，次に述べるプログラム作成の支援の知識と同じデータにする．

４．プログラム作成の支援

4-1 プログラム作成

プログラム作成過程において，関数の使用方法，基礎的なプログラミング技術などが不明確な場合がある．このときに関数の使用方法，関数の使用例などの知識の提示をおこなう．ここで，基礎的なプログラミング技術の部分は前述の機能分析の支援の内容と重複する．

4-2 支援方法

4-2-1 インターフェイス

第３章の機能分析の支援でも述べたように，インターフェイスには大きく２種類の方法がある．プログラム作成過程において，選択方式ではユーザの要求に自由度がなくなると考えられるため，インターフェイスの部分は対話形式による自然言語入力を用いることにする．さらに，第３章で述べたルールに関しても，自然言語入力をおこなえるようにする．本システムの全体的なインターフェイスとして，マウス選択及び自然言語入力により機能分析，プログラム作成の支援を同時におこなうようにする．

4-2-2 自然言語による推論方法

本システムの推論の手順は，まず，ユーザの要求より単純な論理の複数個の条件を導出する．次にこれらの条件で規定される関数を推定するという形式である．

関数を推定する方法として，本システムではＫ

J法により同じ様な処理をする関数をまとめ，さらに，その項目について機能ごとにまとめて階層構造図を作成した．この図の一部をを図４(a)に示す．図４(a)の階層構造図の各枝にルールを設けて，要求に含まれる複数個の条件とマッチングして推論をおこなう．例えば，図４(b)で要求例として「RISUTO NO SENTOU YOUSO NO TYUUSHITU」について説明する．この要求の場合，ルール１のレベルのマッチングをおこなう．"RISUTO"の枝にそってノードA1に進む．同じ様にマッチングを繰り返し最終的にノードC1の「ｃａｒ」が要求に対する関数であると推論できる．

次に要求の文章が省略されている場合について述べる．「RISUTO NO YOUSO NO TYUUSHITU」なる例を考える．先程と同じ様に「要素抽出」の項目までは同様に推論できる．図４(a)よりルール３は "SENTOU","SENTOUIGAI","SAIGO","TORINOZOKU" の４つの枝に別れる．これより，省略されている条件は上記の４つであると考え，４つのルールの選択により推論を進める．この２段階の推論方法により自然言語による推論をおこなっている．また，自然言語では同音異義語について考える必要があるが，本システムでは対象知識をＬｉｓｐ言語に限定したため，考える必要はない．さらに，各ルールの同義語については，Ｌｉｓｐの属性関数を用いてルールデータベースに用意された同義語を語群として登録する．推論をおこなう際，語群より要求の各条件とのマッチングをとるようにする．

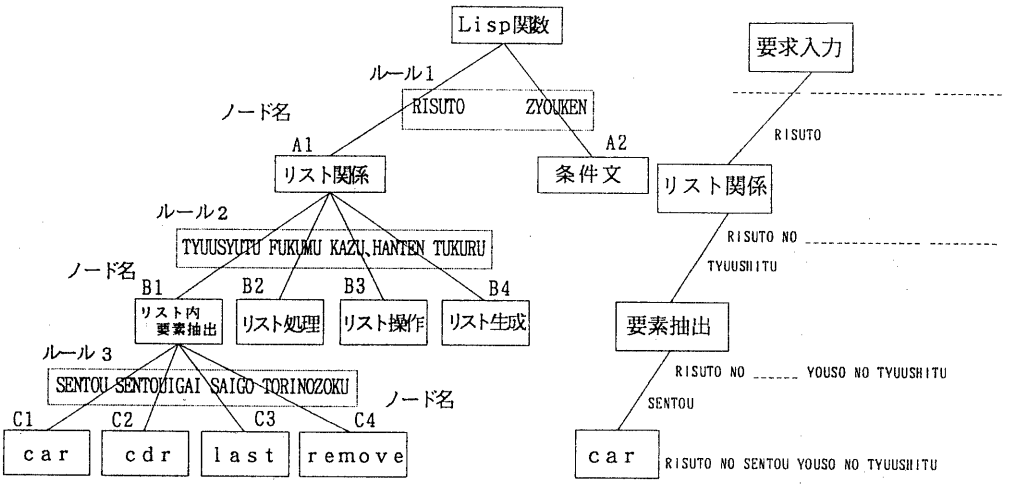

図４　Ｌｉｓｐ関数の階層構造図

5．支援例

本システムの具体的な支援について説明する．

問題例として，「住所録を作り，任意のデータをいつでも表示できるプログラムを作成する．」について説明する．

①本システムの起動（図５）

エディタ上でプログラミング中にウインドウが開き，メイン画面と機能分析を行うためのサブ画面が表示される．

②マウスによりルールの選択

問題の一部として，「任意の人物の住所を取　り出す」について考える．

①で表示されたサブ画面より，必要な処理を選択する．マウスにより「文字列の操作」の部分を選択する．

③図３の構造図より，階層の項目をサブ画面に表示（図６）

②で選択したルールの階層の２項目を表示する．さらに，選択を続ける．マウスにより「情報を扱う」を選択する．

④ユーザの要求に対する知識の提示（図７）

推論を繰り返し，最終的にユーザの要求に対する知識は「探索」であると推論する．マウスで「探索」を選択し，知識の提示をおこなう．

⑤事例，関連する関数，関連するプログラミング技術の選択（図８）

さらに具体的な例（事例），関連する関数や関連するプログラミング技術の知識を必要とす　る場合，選択の画面よりマウスにより選択す　る．

⑥プログラム作成の支援（図９）

ユーザがプログラミング技術の知識を獲得して，プログラミングを始めようとしたとき，関数の作り方がわ

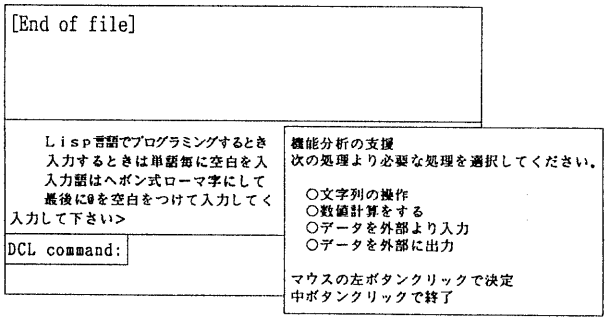

図５　起動画面

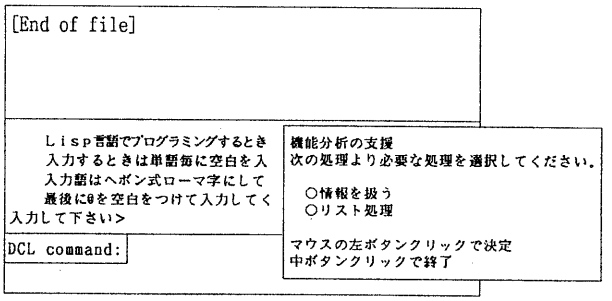

図６　機能分析の支援画面１

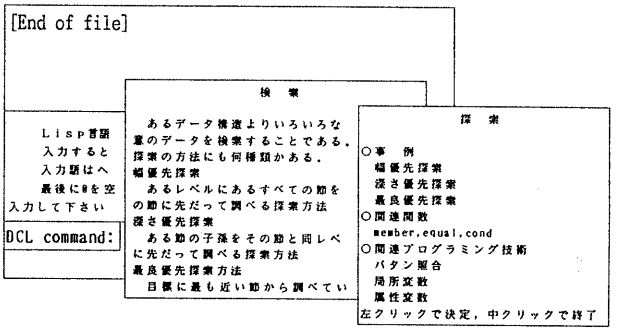

図７　機能分析の支援画面２

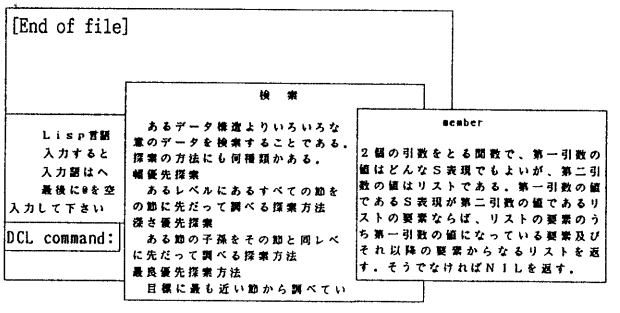

図８　機能分析の支援画面３

からなくなった場合，「KANSUU NO SAKUSEI @」といった要求を入力する．第４章プログラム作成の支援の推論より，ウインドウに「ｄｅｆｕｎ」の説明を提示する．

このような繰り返しで，問題処理の支援をおこなう．

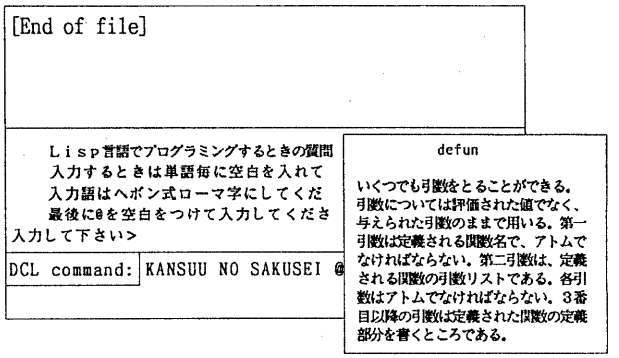

図９　プログラム作成の支援画面

６．おわりに

本研究では，プログラミングの経験の少ない初級者がプログラミングする場合，プログラム作成の知識の提示だけの支援では不充分であり，プログラム作成をする前段階としての，機能分析の支援をおこなう必要があると考え，機能分析とプログラム作成の両方の支援が行えるインストラクションシステムを構築した．機能分析の支援をおこなうためには，ユーザが言語についてある程度わかっていることを前提として，機能ごとのルールを設けて，対話形式により推論が可能である．さらに，プログラム作成の支援として関数の使用方法の知識を用意することにより，機能分析からプログラミングをするまでの支援をおこなうことができた．

今後の課題として，本システムはプロトタイプの開発がすんだ段階である．これより，何人かのユーザに本システムを使用してもらい，評価をする必要がある．さらに，評価によってわかった問題点について検討する必要がある．

７．参考文献

1)上原・山本・小川：「ＬＩＳＰ－ＰＡＬ：プログラミング支援のための自然言語による質問応答システム」情報処理学会論文誌 Vol.30 No.11 (1989)

2)河合・溝口・願化・喜納・角所・豊田：「Prologプログラミング教育のための知的CAIシステムの開発」情報処理学会論文誌 Vol.28 No.3(1987)

3)太田・江藤・丹羽・松田：「Ｃｏｍｍｏｎ　Ｌｉｓｐインストラクションシステム　－インターフェイスを中心として－」第３４回自動制御連合講演会(1991.11.21)

4)J.R.アンダーソン・A.T.コーベット・B.J.ライザー（著），玉井浩（訳）：「Information & Computing = 30　これがＬＩＳＰだ！ -ESSENTIAL LISP-」サイエンス社(1989.2.25)

5)P.H.ウィンストン・B.K.P.ホーン・白井良明（原著）、安部憲広・井田昌之（訳）：「情報処理　シリーズ4-1　ＬＩＳＰ(I)（原書第３版）」培風館(1991.12.10)