ロボット言語について

真泉 史夫

今年の3月にデトロイトで開かれた第6回ロボット展で世界最大のコンピュータ会社であるIBM社が言語をもった2種類の産業用ロボットを発表して注目を集めた．従来の産業用ロボットは，Danger(危険)，Dirty(汚れ)，Dullness(単調)という場面向きのものが多かったが，ここで発表されたロボットは組立作業というDelicacy(精巧)な場面に向いたものであり，1つは日本の三協精機製作所で作られたロボットで他の1つはRSIとよばれるIBM社の試作ロボットである．

この発表が注目を集めたのは，自社開発至上主義を通してきたIBM社が日本製のロボットを採用したこと以上に，「ロボットプログラミング用にALGOL，APLおよびLISPの良い点をひろい集めた」[1] AML (A Manufacturing Language) という名のロボット言語を大きなセールスポイントとしてロボット市場に参入してきたことにある．

本文では，このようなロボット言語の現状について簡単にふれたのちにロボット言語システムの構成要件を概説する．

1. ロボット言語の現状

フレキシビリティに富む生産プロセスにロボットを使うためには，流れてくる品物の変化に応じてロボットの動きを変えてゆく必要があり，そこから「使いやすい言語」の要請がでてきている[2]．

1.1 ティーチングとロボット言語

ロボットに動作の指示を与える方法として現在最も使用されているのはティーチングとよばれる動作点を1つずつ教える方法であり，単純な作業の場合には直観的で習得しやすいという利点がある．しかしロボット利用場面が拡大しかつ周辺機器との連動などの高度化にともない，作業の指示者である人間とロボットおよび周辺機器とのコミュニケーションもより複雑となり，人手を多く取るティーチングに代わる手段としてロボット言語の実用化が求められており，ロボット言語を用意することがロボット市場に参入するための必須事項となりつつある．

1.2 研究開発の状況

ロボット言語の研究は，1960年代後半から人工知能研究の一環としてアメリカのスタンフォード大学やMITおよびイギリスのエジンバラ大学で始まった．特に，スタンフォード大学人工知能研究所で開発されたロボット言語ALは，ロボットの動作ならびに対象物記述の抽象化に本格的に取り組み，今日のロボット言語研究の源流となっている[3]．

アメリカでは，現在ユニメーション社のVAL[4]，オートマティックス社のRAIL[5]，マクドネルダグラス社のMCL[6]などのロボット言語が開発されている．一方，ヨーロッパでは，イギリスのエジンバラ大学のRAPT[7]，フランスのLM[8]，ドイツのROBEX[9]，イタリアの

まいずみ ふみお ㈱ジェー・エム・エー・システムズ システム計画部

MAL [10] などが研究段階から実用化に向かいつつあり，ロボット言語の標準化などをテーマとした研究者間のコミュニケーションもさかんである．

日本では，小松製作所が第11回国際産業用ロボットシンポジウムで溶接作業用言語PLAWの発表を行なっている[11]．また山梨大学の牧野を中心としたSCARA研究会[12]に参画したロボットメーカーが商品化している産業用ロボットの多くは，三協精機のSERF [12] や日本電気のPASLA [13] などのようにロボット言語をつけて市場に出されている．

これに加えて，前に述べたIBMの動向に見られるように内外のコンピュータメーカーが，そのソフトウェア開発力をロボット言語にも向けつつあり，ロボット言語の研究開発とその実用化に加速度がついてきている．

1.3 ロボット言語の役割

ロボット言語の役割は，一般のプログラミング言語の役割と相対で考えると次のようにとらえられる[14]．

(1) ロボットに動作を指示する手段．

(2) ロボット利用の技法を広めるための技術交流の媒体．

(3) その技術を蓄積するライブラリのための媒体．

さらに言語一般の性格として，

(4) ロボット利用上の思考の媒体．

(5) 概念形成上の培養基．

ともなるといえる．

2. ロボット言語システムの構成要件

ここまで「ロボット言語」とよんできたものはより正確には「ロボット言語システム」とよぶべきであり，図1に示すようにロボット本体，対象物，および周辺機器を対象として，作業指示や動作指示を与える言語そのものと，それを解読しロボットを操作する処理系，ならびにロボットの世界に関する情報をもつ環境モデルという3つの要件から構成されている．

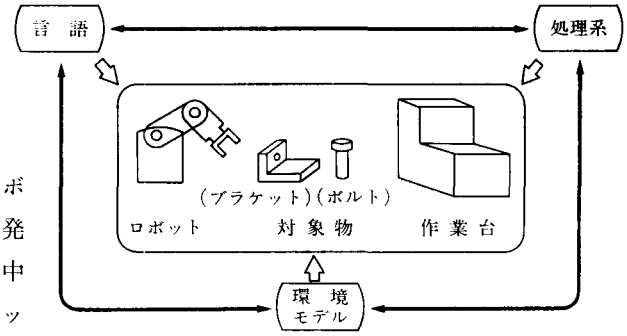

図 1 ロボット言語システムの構成要件

2.1 ロボット言語の定義要素

ロボットを動かすには次の情報が必要となる．

◎環境情報……ロボットの動く環境の記述．これは，ロボットの世界を表現したもので，ここで与えられた情報だけによってロボットは判断を行なうことになる．

◎動作制御情報……ロボットの動きとそれに付随する種々の条件，他のロボットとの関係，動作の行ない方，精度などの情報の記述．

これらの2つ情報の取扱いが可能なことがロボット言語に最低限必要なことといえよう[15]．

この条件を満たすためのロボット言語の定義要素は，ALを参考にして整理すると次のようになる[16]．

(1) 環境データ定義

ロボット自体，対象物，作業台，部品供給装置搬入出装置，障害物などのロボット世界に存在する物体の形状とその位置・方向の定義．

(2) 対象物の組付状態の定義

対象物同士が組み付けられたり取り去られたりすることの定義で，これにより各対象物に対して与えられた位置と方向が固定されたり分離されたりする．

(3) 動作指示の定義

ロボットの動きの定義であり，各関節へ直接指令する方法と対象物の移動を記述することにより

```
ブラケット←FRAME(ROT(Z, 90*DEG), VECTOR(20, 40, 0));
                                                (環境データ定義)
ブラケット握点←ブラケット*FRAME(ROT(X, 180*DEG), VECTOR(0,1,5));
                                                (同 上)
ブラケットの穴←ブラケット*FRAME(ROT(X, 180*DEG), VECTOR(5,2,0));
                                                (同 上)
      ⋮
FOR I←1 UNTIL 3 DO      次の処理を3回繰返す.(制御文の定義)
    BEGIN               開始                    (同 上)
    CENTER      右腕:   右腕を中心にすえる. (動作指示の定義)
    ブラケット握点←右腕:  右腕でブラケット握点を把持.(同 上)
    AFFIX ブラケット TO 右腕 RIGIDLY:  ブラケットと右腕が結合状態.
                                        (対象物の組付状態の定義)
    MOVE ブラケット TO 作業台
        WITH APPROACH=2*INCHES:  ブラケットを作業台から2インチ
                                 のところに移動.(動作指示の定義)
    OPEN 右手 TO 3.5*INCHES:        右手を3.5インチ開く.(同上)
    UNFIX ブラケット FROM 右腕:  ブラケットが右腕からはなされた状態.
                                 (対象物の組付状態の定義)
      ⋮
INSERT ボルト TO ブラケットの穴   ブラケットの穴にボルトを挿入.
                                 (より強力な高級定義)
        WITH FORCE=5000*DYNES:  ちからは, 5000ダイン.
                                 (知的機能の付加記述)
      ⋮
```

図 2 ロボット言語ALによる記述例

ロボットの動きを指定する方法とがある.

(4) 知的機能の付加記述

時間，温度，力などの動作条件の定義でセンサーによるモニタリング機能を前提としている.

小松製作所がアーク溶接用に開発した PLAW は，触覚センサーをたくみに組み入れてできている．また視覚センサーを扱ったものとしてはアメリカのオートマティックス社の RAIL がある.

(5) 制御文の定義

繰返しや条件分岐のようなプログラムの流れを制御する定義で，構造化プログラム技法の考慮が必要である.

(6) コンパイル時の定義

条件により異なる部分をコンパイルさせる定義で，必ずしも必要ではない.

(7) より強力な高級定義

類似した手順指定のうちの共通項を標準文節とし，変化する可能性のある項をパラメータとして代入することにより，ひとまとまりの作業を定義する方法.

図2に，以上のような定義要素を使ったロボット言語の記述例を示す.

2.2 ロボット言語のレベル

ロボットの動作は手先の動きというミクロなレベルから組立作業というマクロなレベルまでありこれを一様な言語であつかうには無理がある.

そこで，たとえば AL では下位言語と上位言語という2つのレベルを設定し，この問題を解決しようとしている[17].

◎下位言語……腕の運動や手首の動作などのロボットハードウェアの動作に対応する動作指示のための言語.

◎上位言語……ポンプヘッドの組付けとかスタッドの挿入のような工程要素作業を指示できる言語.

このように下位言語が，プログラムをロボット制御用のハードウェアのコードから切り離すことを目的としているのに対し，上位言語は，プログラムをロボットの単位動作の指示から切り離すことを目的としている.

最近のロボット言語研究では，図3に示すように AL でいう下位言語を動作指示言語，上位言語を作業指示言語とよび，それぞれの目的に適合した言語のあり方が追求されている.

また，フランスのロボット言語 LM の開発者は機械組立の分野ではロボット言語を図4に示すような2つの概念レベルに分類できるとしている.

「Effector Level」

組立作業をロボットの手先 (End Effector) の操作と，センサーによって与えられるデータという2つの言葉で定義する概念.

「Object Level」

ロボットの操作とは独立に部品間の組立関係という言葉で組立作業を定義する概念.

現在，開発されている言語のほとんどは前者に属するが，イギリスのエジンバラ大学で開発されている RAPT は，Object Level のアプローチをとっている.

日本産業用ロボット工業会のロボット言語研究

指示レベル	ロボットの動き	記　述　例
動作指示	はめ合せ 軸合せ 出入口 接近点 かつ退避点	ENTRY 作業空間に入る. APPROACH 接近点に行く. ALIGN 軸を合わせる. ENGAGE はめ合わせる. OPEN 手先を開く. DEPARETURE 退避点にもどる. EXIT 作業空間から出る.
作業指示		INSERT ボルト INTO ブラケット穴 USING 治具-1 治具1を使ってブラケットの穴にボルトを挿入する.

図 3　動作指示と作業指示

図 4　概念レベルによる分類

概念レベル	ロボットの動き	記　述　例
Effector Level		GRASPボルトFROMパレット INSERTボルトTOボックスの穴
Object Level		FITS/ボルト，ボックスの穴

で設計された作業指示向き言語の中でも，このアプローチが部分的に試みられている[18].

そこでは，「組立対象部品は必ず目標に対してはめ合わせの部位(ESP：Engage Sub Part)をもち，同時にその組付がロボットの手先によって行なわれることを前提として手先によってつかまれる部位(GSP：Grasp Sub Part)がある．次にその部品が組み付けられた後，他の部品をその上にはめ合わせることになると相手とはめ合わされる部位(TSP：Target Sub Part)が1つ以上ありうることになる」としている.

このように，ある部品の ESP は他の部品の TSP とつながり，自身のもつ TSP は別の部品の ESP とつながり，全体の組立製品構成の関係が表わせる．この情報が，ロボット動作の生成に大きな役割をもつとされている.

2.3　処理系の構成

ロボット言語で記述されたプログラムを解釈しロボットおよび周辺機器を操作する処理系は，それぞれの言語システムが扱う機能範囲や想定するコンピュータのレベルにより一様ではないが，ひとつのあり方として図5に示すようなコンポーネントからなる処理系が考えられる[19].

(1) コンパイラ

ロボット言語ソースプログラムを読みマクロ展開し，動作のシミュレートを行ない，オブジェクトコードを出力する.

(2) 環境モデルマネージャ

コンパイラから動作文をうけとり，環境モデルを参照して対象の座標や握り点などを計算し，ディフォルト値をおぎない，中間コードを作りコンパイラに返す.

(3) 軌道計算器

オブジェクトコードの中の軌道情報にしたがって実際のサーボ制御用の軌道計算や軌道データを作る.

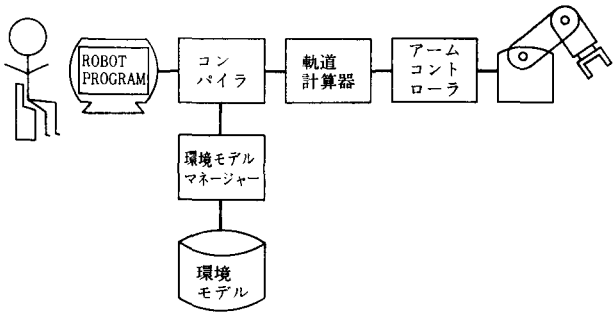

図 5 ロボット言語の処理系

(4) アームコントローラ

オブジェクトコードと軌道データの両方をうけとり，サーボ制御を実行する．

さらにこのほかに，会話方式でプログラムを作り，ロボットの操作を行なうためのインタプリタやデバッグ用ツールなどのコンポーネントがありうる．

2.4 処理系の使いやすさと移植可能性

ロボット言語システムを使って作業のプログラミングを行なう場合には，次の4つのフェーズが必要となる．

⊙プログラム書き……プログラムの最初の版を作る．

⊙初期設定……アームの位置づけ，パーツフィーダマガジンの取付け，治具の初期化，道具の交換，新しい作業対象物のセッティングなどを行なう非常に煩雑なフェーズ．

⊙テスト……命令を実行し，誤りを発見する．

⊙修正……文を修正し，再コンパイルし，中断点からテストを再開する．

これらの各フェーズごとの使いやすさの配慮が処理系としてなされていることが現場にロボット言語を普及させるためのポイントの1つとなる．

一方，現在のロボット言語システムの処理系の多くは，特定のロボットとコンピュータを対象としているために移植が困難なつくりとなっていてこれがロボット利用技術の交流の妨げとなっているという問題がある．

処理系を移植可能とするには，そのつくりをロボットから独立させ，さらにコンピュータからも独立させる必要がある．このためには次のような点の考慮が必要となる．

⊙ロボットからの独立

NCなどと同様に，中間コードの標準化を行なう．

⊙コンピュータからの独立

処理系を記述する言語として標準プログラミング言語を使う．

2.5 環境モデル

ロボットの世界に存在する物体の形状とその位置方向および物体間の関係に関する情報を分析して使用しやすい形式にし，かつ定義間の関係を明確に表現した形式で統一的に取り扱うのが環境モデルである．この概念は，ロボットを含めたCAD/CAM領域では一般にGeometric Modelingとよばれている[20]．

図6に示すような環境モデルをもった言語システムは，IBM社のAUTOPASS[21]や東京大学のGEOMAPを利用した言語システムの試作にみられる程度である[22]．大半のシステムは環境モデルが非常に複雑で大型コンピュータを必要とすることをきらい，処理系内部で必要最少限の環境情報だけを扱っている．

電総研では，環境情報の初期登録を容易にするためにレーザーポインタを用いたロボットの環境教示システムを開発した[23]．

今後は，このような援助を背景として環境モデルをもつロボット言語システムがかなり増えてく

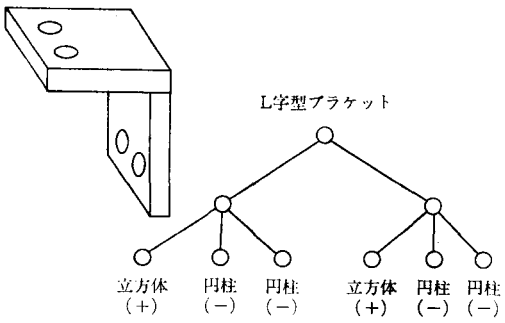

図 6 環境モデル内の部品形状表現

ると思われる.

以上，ロボット言語の現状とロボット言語システムの構成要件について述べてきた.

各ロボット言語の特色については，(社)日本産業用ロボット工業会のロボット言語分科会でまとめたものを次ページ*にのせてあるので参考としていただきたい.（*出所：(社)日本産業用ロボット工業会）

今後のロボット言語の研究開発が，現状の多種多様なロボット言語を標準化し，ロボット利用技術の交流が広く行なわれるような基盤を作る方向に進むことを望む.

参考文献

[1] Tom Manuel : Personal Computer can program high-assembly robot introduced by IBM, *Electronics*, March 10 (1982), 42-44

[2] 編集部：ロボット事業化のカギはセンサーとソフトを制することだ，マネジメント，3月号(1982), 36-39

[3] S. Mujtaba, Ron Goldman : AL User Manual, Stanford Artificial Intelligence Lab., (1977)

[4] 窪田八洲洋：PUMA シリーズにおける言語システム，オートメーション，Vol. 27 No. 4. (1982) 44-47

[5] 丸地三郎：実用化時代に入った視覚「オートビジョンII」,オートメーション，Vol. 27 No. 2 (1982) 76-80

[6] M. A. Eastwood : Robot Programming Languages, Proc. CAM 8th Annual Meeting, (1979) 95-102

[7] R.J. Popplestone, et al. : RAPT A language for describing assemblies, *The Industrial Robot*, Sep. (1978) 131-137

[8] J.C. Latombe, E. Mazer : LM A High-Level Programming Language for Controlling Assembly Robots, Proceedings of the 11th International Symposium on Industrial Robots (Proc. 11th ISIR), (1981), 683-690

[9] M. Weck : ROBEX An Off Line Programming System for Industrial Robots, Proc. 11th ISIR, (1981), 655-662

[10] G. Gimi, et al. : Introducing Software Systems in Industrial Robots, Proc. 9th ISIR, (1979), 309-321

[11] H. Takagi : The Programming Language for Welding Robot, Proc. 11th ISIR, (1981), 691-698

[12] 安川員仁，林善雄：SKILAMにおける言語システム，オートメーション，Vol. 27 No. 4, (1982) 58-64

[13] 諸田昇，岩崎真人：「NEBOT」の開発と応用，ロボット，No. 32 October (1981), 17-23

[14] 竹下亨：日本におけるプログラミング言語，bit, プログラミング言語特集号，(1974), 264-270

[15] 平山慎一郎他：ロボット言語に関する基礎調査研究，(社)日本産業用ロボット工業会，エンジニアリングプロジェクト調査研究報告書，(1977), 26-171

[16] 真泉史夫：ロボットの制御システムと言語，オートメーション，Vol. 27 No. 4, (1982), 18-26

[17] 小野勝章：ロボット言語について，ロボット，No. 21 (1978), 10-16

[18] 平山慎一郎他：利用者向き組立作業指示言語，(社)日本産業用ロボット工業会，エンジニアリングプロジェクト調査研究報告書，(1980), 46-117

[19] H. Inoue, et al. : Design and Implementation of High Level Robot Language, Proc. 11th ISIR, (1981) 675-682

[20] 沖野教郎，久保洋：形状モデリングと CAD/CAM，情報処理，Vol.21 No.7, (1980), 725-733

[21] L.T. Lieberman, M.A. Wesley: AUTOPASS An Automatic Programming System for Computer Controlled Mechanical Assembly, *IBM J. RES. DEV*, No. 21, (1977)

[22] 佐田登志夫，木村文彦：幾何モデルによる工業用ロボット作業プログラミングシステムの開発，昭和56年精機学会春期大会論文集，592-594

[23] 石井優：レーザを用いた物体認識と環境教示システム，オートメーション，Vol. 27 No. 2 (1982), 23-29

No.	1	2
名　称	AL 〔Assembly Language〕	AUTOPASS 〔AUTOmated Parts Assembly system〕
開発元 (国)	Stanford Artificial Intelligence Laboratory (米　国)	IBM Watson Research Laboratories (米　国)
出　典	AL USERS' MANUAL Nov. 1977	IBM J.Res. Develop., July 1977
概 要	後の研究に影響を与えた多くの概念を作り出した．計画時システムと実行時システムを分離しており，また物体記述をデータベース化しようとしている．ユーザーが与えたworldモデルをもとに，あらかじめ軌道を計算する． 上位言語設定の機能があり，ハードウェアの動作と作業指示との独立をめざしている． PDP-10，PDP-11でランするALGOLライクな言語である．	組立のための細かなロボット指令をなくすことを目的としている．ユーザーではなく，AUTOPASSが把握点，軌道，および部品を組み立てるのに必要な動作を選定する．部品モデルの幾何データベースおよび大型コンピュータの処理能力のおかげでAUTOPASSは，他の言語より自動化されているといえるが，今のところ，まだ実験段階である． PL/Iライクな言語である．
記 述 例	BEGIN "insert peg into hole" FRAME peg_bottom, peg_grasp, hole_bottom, hole_top; MOVE barm TO bpark WITH DURATION=3*seconds; OPEN bhand to 3*inches;　{ normal initialization } peg_bottom ← FRAME(nilroln,VECTOR(20,30,0)*inches); hole_bottom ← FRAME(nilroln,VECTOR(25,35,0)*inches); AFFIX peg_grasp TO peg_bottom RIGIDLY AT TRANS(ROT(xhat,180*degrees),3*zhat*inches); AFFIX hole_top TO hole_bottom RIGIDLY AT TRANS(nilroln,3*zhat*inches); MOVE barm TO peg_grasp; CENTER barm;　{ get peg } AFFIX peg_grasp TO barm RIGIDLY; MOVE peg_bottom TO hole_top; MOVE peg_bottom TO hole_bottom DIRECTLY { To ensure that arm does not lift up and go down again} WITH FORCE_FRAME = station IN WORLD WITH FORCE(zhat) = -10*ounces WITH FORCE(xhat) = 0*ounces WITH FORCE(yhat) = 0*ounces SLOWLY; END "insert peg into hole";	1. OPERATE *nutfeeder* WITH *car-ret-tab-nut* AT *fixture.nest* 2. PLACE *bracket* IN *fixture* SUCH THAT *bracket.bottom* CONTACTS *car-ret-tab-nut.top* AND *bracket.hole* IS ALIGNED WITH *fixture.nest* 3. PLACE *interlock* ON *bracket* SUCH THAT *interlock.hole* IS ALIGNED WITH *bracket.hole* AND *interlock.base* CONTACTS *bracket.top* 4. DRIVE IN *car-ret-intlk-stud* INTO *car-ret-tab-nut* AT *interlock.hole* SUCH THAT TORQUE IS EQ 12.0 IN-LBS USING *air-driver* ATTACHING *bracket* AND *interlock* 5. NAME *bracket interlock car-ret-intlk-stud car-ret-tab-nut* ASSEMBLY *support-bracket*

No.	3	4
名　称	DIAL 〔Draper Industrial Assembly Language〕	INDA 〔INDustrial Automation〕
開発元 (国)	Charles Stark Draper Laboratory (米　国)	SRI International (米国) Philips Research Laboratories (英国)
出　典	Proceedings 1979 CAM-I International Spring Seminor	Proceedings 9th ISIR, March 1979
概 要	プログラミングの容易さをめざした言語であり，4つのレベルのプログラマの階層をもつシステムである．すなわち，システム・プログラマ，戦略プログラマ，作業手順プログラマ，そして作業現場オペレータ/プログラマである．本研究では，Data General Nova 2ミニコン上で，RDOSマルチタスクを使って実行されている．DIALは，コンパイラーインタープリータ型の言語であり，ソース・プログラムは，ML/Iとよばれるマクロ・プロセッサによって，コンパクトなオブジェクト・コードへと翻訳される．マクロ定義機能は，強力である．しかし，効率が悪いという欠点がある．	このシステムは，RTL/2で書かれており，Philips P857 32K words, 16bitミニコン上で，Philipsのディスク・オペレーティング・システムDOMを用いてランする． 開発のフェーズに対し，コンパイラ，インタープリータ，エディタおよびデバッグ用ツールを組み合わせ，使いやすさを重視した処理系を実現している．

記
述
例
	MCDEF<COLLETPICKUP> AS<MOVE (11, UC, 2; 3, CG)
	MOVE (1, LC, 4)>
	
	MCDEF<COLLETINSERT> AS<MOVE (5, UC, 6, 7, CR)
	MOVE (8, LC, 9)>
	
	CLEAR TOOL
	LOCK
	MOVE (1)
	COLLETPICKUP
	COLLETINSERT
	COLLETPICKUP
	COLLETINSERT
	COLLETPICKUP
	COLLETINSERT
	COLLETPICKUP
	COLLETINSERT
	MOVE (1)
		LINE　TEXT　START
		120 IF NPARTS < 12 THEN ¦ 120
		130 PICK (12 - NPARTS) ; ¦ 130
		140 ELSE ¦ 120
		150 PUT (NPARTS) ; ¦ 150
		160 END ; ¦ 120

No.	5	6
名　称	LAMA-S	LM 〔Language for Manipulation〕
開発元 (国)	PROJECT SPARTACUS, IRIA (フランス)	Artificial Intelligence Group at IMAG (フランス)
出　典	The Industrial Robot, September 1980	Proceedings 11th ISIR, October 1981
概 要	新しいロボット言語，およびプログラミング法を開発するためのリサーチ・ツールとして提案されたものである．APLで書かれており，基本的には，下位言語であるPRIMAを生成する，拡張性のあるコンパイラとして使用される．またプログラムを並行に実行させることができる． センサー入力が可能であり，マクロ命令を定義することにより，PLACE, CHANGE, DETECTなどの作業指示も可能となっている．	一般的シンタックスは，PASCALライクな言語である．ALと同様，フレームだけを扱い，三次元実体モデルをもっていないので，プログラマが衝突チェックを行なわなければならない．ティーチング・プレイバック機能をもち，センサー入力が可能である． LSI11/3(64KB×16bit)マイコンを使用し，FORTRANとアセンブラで書かれている．
記 述 例	∇R ← A IN B R ← A, B ∇ ∇R ← C AT D R ← O, C, D ∇ ∇R ← E VIA F X ← F[0] + 1 F ← 1 ↓F R ← X, E, F ∇ ∇ CHANGE G . . . . READP G[1] ∇ Returns the addresses of <destination> and <time> Returns a counter initialised at O and the addresses of <pointn>, <destination> and <time> Returns after a number of calls, the number of intermediate points, the addresses of these points and the addresses of <destination> and <time> Execution of the MOVEs through the intermediate points Update the new location of the object	PROCEDURE INSERT(PIN FRAME, HOLE FRAME, FZMAX REAL, EPS REAL, NMAX INTEGER) BOOLEAN; INTEGER N; VECTOR LATF; FRAME HOLE1; BEGIN N:=0; HOLE1:=HOLE*TRANSLATION(-VZ,1D); WHILE N<NMAX DO BEGIN MOVE PIN BY TRANSF(PIN,HOLE1) UNTIL FZ(1)>FZMAX /; IF DISTANCE(PIN,HOLE)<1 THEN RETURN(TRUE) ELSE BEGIN LATF:=VECTOR(FX(1),FY(1),0); MOVE PIN BY TRANSLATION(LATF,EPS) /; EPS:=EPS/2; N:=N+1; END END; WRITE "INSERTION FAILED"; RETURN(FALSE) END

No.	7	8
名　称	MAL 〔Multipurpose Assembly Language〕	MCL 〔Manufacturing Control Language〕
開発元 （国）	Milan Polytechnic （イタリア）	McDonnell Douglas Corporation （米　国）
出　典	Proceedings 9th ISIR, March 1979	ROBOT PROGRAMMING LANGUAGES
概 要	簡単で低価格のプログラマブル・システムをめざして，設計された言語である．組立作業用に主として設計された，2本のアームをもつロボットSupersigma robotのプログラミングおよびコントロールのための完成されたソフトウェア・システムである． BASICライクな言語で，2本以上のアームをもつロボットの異なる作業を別々にプログラミングすることができる． MALシステムは，FORTRANで記述されており，20K words（16bit）で十分であり，24K memory wordsのイタリア製ミニコン上でランする．	さまざまなアプリケーションに適用でき，複数ロボットの協調作業も行なえる．次のような特徴がある． ①ロボットだけでなく，N/C機械，センサー，カメラ，そして，これらの調整用コンピュータのための総合製造用言語である． ②あらゆるタイプのロボットに使用できる． ③視覚操作の中のマニュアルによる操作を少なくできる．
記 述 例	5　SET IP=10, MP=150.5, FP=700, TABLE=15.7, UP=TABLE+40 6　SET LEFTHAND=8 7 SET YES = 0, NO = 1, OBJECTREADY = NO 8 TASK 2, 150 9 "---TASK NO. 1 - LEFT ARM --- 10　MOVE W ZL = UP 15 SET FREEAREA = YES 20　MOVE W XL = IP 30　MOVE W ZL = TABLE 40　ACT LEFTHAND　"close left hand 50　MOVE W ZL = UP 55 WAIT FREEAREA　"may I access my area? 57 SET FREEAREA = NO　"yes. It's my own now! 60　MOVE W XL = MP 70　MOVE W ZL = TABLE 80　DEACT LEFTHAND　"open left hand 85 SET OBJECTREADY　"object is now in MP 90　GO TO 10 100　SET RIGHTHAND = 9 109 " --- TASK NO. 2 - RIGHT ARM --- 110　MOVE W ZR = UP 115 SET FREEAREA = YES 120　MOVE W XR = FP 130　MOVE W ZR = TABLE 140　DEACT RIGHTHAND　"open right hand 150　MOVE W ZR = UP 155 WAIT OBJECTREADY, FREEAREA　"waiting for an object in MP 157 SET FREEAREA = NO 160　MOVE W XR = MP 170　MOVE W ZR = TABLE 180　ACT RIGHTHAND　"close right hand 185 SET OBJECTREADY = NO　"no object in MP now 190　GO TO 110	$$ CHECK THE SIZE OF A CUTOUT DEVICE/CAMRA1 INSPEC/PROJEC,PARTA,REGION,CUTOUT,AREA,20,STATUS$,RESULT WHEN (RESULT.EQ.OK) 　DEVICE/ROBOT2 　GOTO/GOODPT ELSE 　GOTO/BADPT

No.	9	10
名　称	RAIL 〔Robot Automatix Inc. Language〕	RAPT 〔Robot APT〕
開発元 （国）	Automatix Inc. （米　国）	University of Edinburgh （英　国）
出　典	オートメーション，1982年2月号	The Industrial Robot, September 1978
概 要	わかりやすく，作成しやすい点が長所であり，コンピュータに慣れていない人にも短期間で覚えて使いこなすことができる．会話型の実行が可能であり，1つ1つの命令のつど，実行することができる． 視覚センサーをとり入れた言語で，部品検査への応用が行なわれている．	組立用ロボット制御言語であり，DEC10コンピュータ上で実験的にインプリメントされている．組立用ロボットは，NC工作機械と似ているので，NCに慣れた人が，すばやく適応できるように文法の基本にAPT言語を使用している．マクロ機能も用いられている． また，組立作業をロボットの操作とは独立に部品の組立関係という言葉で定義しようというObject level のアプローチをとっている． 現在は，小さなアプリケーションでも，DEC-10上でランするのに数分かかってしまうので，使用可能なインタープリータにして，DEC-11あるいは他の16ビットマシン上でインプリメントすることが目標である．

記
述
例
	INPUT PORT 1 : CONVEYOR
	INPUT PORT 2 : PART DETECTOR
	OUTPUT PORT 1 : BAD PART
	WRITE ENTER CHIP OFFSET TOLERANCE :
	READ OFFSET TOL
	WRITE ENTER CHIP TILT TOLERANCE :
	READ TILT TOL
	WAIT UNTIL CONVEYOR=ON
	WHILE CONVEYOR=ON DO
	　BEGIN
	　　WAIT UNITL PART DE TECTOR=ON
	　PICTURE
	　IF XMAX "HEAT SINK" XMAX "CHIP"=OFFSET
	　TOL AND
	　　ORIENT "CHIP" WITHIN TILT TOL OF 90
	　　THEN
	　　　BAD PART OFF
	　　ELSE
	　　　BAD PART ON
	END
		BLOCK = MACRO/B X Y Z R;
		　BODY/B;
		　　REMARK THE RELEVANT FEATURES OF THIS STANDARD BLOCK WILL BE THE TOP, THE BOTTOM, THE BACK, TWO SIDES AND TWO HOLES DRILLED THROUGH THEM:
		　　P1=POINT/0,0,0;
		　　P2=POINT/X,0,0;
		　　P3=POINT/0,Y,0;
		　　P4=POINT/0,0,Z;
		　　P5=POINT/ X/4,Y/2,0;
		　　P6=POINT/ X-X/4,Y/2,0;
		　　C1=CIRCLE/CENTER,P5,RADIUS,R;
		　　C2=CIRCLE/CENTER,P6,RADIUS,R;
		
		　　L1=LINE/P1,P2;
		　　L2=LINE/P1,P3;
		　　L3=LINE/P3,PARLEL,L1;
		　　L4=LINE/P2,PARLEL,L2;
		
		　　BACK1=FACE/L2,XSMALL;
		　　REMARK THE BACK OF THE BLOCK
		　　BOT1=FACE/HORIZONTAL,0,ZSMALL;
		　　REMARK THE BOTTOM;
		　　TOP1=FACE/HORIZONTAL,Z,ZLARGE;
		　　REMARK THE TOP;
		　　RSIDE1=FACE/L1,YSMALL;
		　　REMARK THE RIGHT SIDE;
		　　LSIDE1=FACE/L3,YLARGE;
		　　REMARK THE LEFT SIDE;
		　　HOLE1=HOLE/C1,ZLARGE;
		　　HOLE2=HOLE/C2,ZLARGE
		
		　TERBOD;
		TERMAC;

No.	11	12
名　称	ROBEX 〔ROBoter EXapt〕	RPL 〔Robot Programming Language〕
開発元 （国）	Machine Tool Laboratory, TH Aachen （西ドイツ）	Stanford Research Institute, International （米　国）
出　典	Proceedings 11th ISIR, October 1981	ROBOT PROGRAMMING LANGUAGES
概 要	APTライクな，EXAPTに似た構造をもつオフラインプログラミング・システムであり，その形状記述機能を利用している．ロボットのハードウェアから独立しており，End-Effectorレベルの言語である． Worldモデルをもち，自動衝突回避による軌道生成，インタラクティブなセンサー入力などを予定している． PDP11/34，PRIME200でランする．	Unimateロボットを操作するプログラムに結合可能な，あらかじめ定義されたルーチン・セグメントのライブラリーに特徴がある．
記 述 例	1 PARTNO/DEMONSTRATION PROGRAM 2 MACHIN/ROBBY 3 $$ 4 $$GEOMETRIC DEFINITIONS 5 $$ 6 MF=MATRIX/XYROT,-180,TRANSL,1240,-1000,600 7 MW1=MATRIX/TRANSL,1300,1400,700 8 MW2=MATRIX/TRANSL,-1300,1400,700 9 MP=MATRIX/XYROT,-180,TRANSL,-500,-500,100 10 TRASYS/MF 11 P1=POINT/250,250,100 12 TRASYS/MW1 13 P2=POINT/0,0,100 14 TRASYS/MP 15 P4=POINT/700,700,100 16 TRASYS/MW2 17 P5=POINT/0,0,100 18 TRASYS/NOMORE 19 P3=POINT/-700,0,500 20 BAFPOS/0,500,800 21 PAL=PALTIZ/LINEAR,P4,-400,-400,0,2,2,2 22 $$ 23 $$PROGRAM START 24 $$ 25 GRIPNO/1 26 OPENGR 27 RAPID 28 GOTO/BAFP 29 $$ 30 $$WAIT LOOP 31 $$ 32 START)ONBIG/EVENT,3,JMP,WIM1 33 ONBIG/EVENT,5,JMP,WIM2 34 JUMP/START 35 $$ 36 $$REQUEST WIM1 37 $$ 38 WIM1)GOTO/P2,EX 39 JUMP/WIMALL 40 $$ 41 $$REQUEST WIM2 42 $$ 43 WIM2)GOTO/P5,EX 44 $$ 45 $$PALETTIZE AND RAW PART PICK UP 46 $$ 47 WIMALL)FEDRAT/300 48 GODLTA/-100 49 CLOSGR 50 GODLTA/100 51 GOTO/P3 52 PUT/PAL,TEACH 53 RAPID 54 GODLTA/300 55 GOTO/P1,EX 56 GODLTA/-100 57 CLOSGR	INVIS　　!INITIALIZE LINK TO VISTON !COMPUTE CAMERA SCALE FACTOR　　MODOLE FDIVR　　(XSCALE, -4096.,25.4) RSETN　　(YSCALE, XSCALE) IFNO　　("DO YOU WANT INSTRUCTIONS", OPEN　　LSNOINST) MOVETO　　PICKUP JOYON　　(02)

No.	13	14
名　称	SIGLA 〔SIGma LAnguage〕	TEACH
開発元 (国)	Olivetti (イタリア)	Bendix Corporation (米　国)
出　典	Proceedings 8th ISIR, 1978	ROBOT PROGRAMMING LANGUAGES
概 要	8K words のミニコン上でランし，マス・メモリーは不要であり，並行作業の可能な言語である．組立て，穴あけ，調整，選別あるいはSIGMAの構成が3～8自由度の1，2，3，4本のアームからなるシステム上にインストールされる． SIGLAの主要機能は，次のとおりである． ①並行作業：多くのアームの独立動作コントロールが可能 ②インタープリータ構造：テレタイプのコントロール下で“teaching by doing”が可能 ③容易に変更できる命令セット：最良のソフトウェア・テーラリングが可能	VALと同様，使用範囲は広い．特に協調動作する2本のマニピュレータの作業に適している． 記述例中のAFTERとNEXTは，協調作業を指示するためのものであり，VISRUNは，カメラ操作を呼び出すものである．
記 述 例	MO - Movement AX - Auxiliaries WA - Wait HL - Hold RP - Hit solid obstacle PP - Part present NU - Number BL - Branch Less BE - Branch Equal BG - Branch Greater JU - Jump EX - Execute AD - Add SE - Set CH - Chaining II - Incremental program (start) IF - Incremental program (end) RI - Reference	140 AFTER 100 POS ARM1 to DSKPO MODE=3 SPEED=.25 150 POS CAMERA1 TO 1 160 NEXT COMPUTE 'VISRUN' FOR MANIP1 AND VIS1 ARG 'FACNUM' = FACENO AND 'SILNUM' = SILHNO 170 SKIP MANIP1 AND VIS1 to 200 IF FACENO>=0

No.	15	16
名　称	VAL	WAVE
開発元 (国)	Unimation, Inc. (米　国)	Stanford Artificial Intelligence Laboratory (米　国)
出　典	USER'S GUIDE TO VAL, February 1979	The Industrial Robot, March 1977
概 要	VALは，長年にわたって使用・改良されてきた言語である．Unimation PUMAロボットを動かすための言語であり，マニピュレータ本体，制御用コンピュータ，周辺機器の3つの要素を統合するものである．DECの16bitマイコンLSI-11/2上でランする．BASICのようなインタープリータ型の言語であり，教示されたデータを取り込んで動作令命に翻訳して記憶することもできる． センサー情報入力や，他のマニピュレータなどの外部機械との連係動作能力もある． VALでは，動作はすべて手先の移動として表現する．	動作を記述し，力および接触コントロールが可能で，また，外部の視覚システムとの結合もできる．インタープリータ型の言語で，PDP-10のTSSの下で動く． 構造は，アセンブリ言語と似ている． アームは，Scheinmanアームを使用しており，T型水ポンプの組み立て，クランク回しなどの仕事をするのに使用された．．
記 述 例	.EDIT DEMO1 .PROGRAM DEMO1 1.?OPEN 2.?APPRO PICK,50 3.?SPEED 30 4.?MOVE PICK1 5.?CLOSEI 6.?DEPART 70 7.?APPROS PLACE,75 8.?SPEED 20 9.?MOVES PLACE 10.?OPENI 11.?DEPART 50 12.?E .	STOP FV, MV ; Stop on 50 oz. MOVE K ; before we reach K SAVE SURFACE ; Save the difference in SURFACE OPEN 5 ; and release the block RESTORE SURFACE ; Move to L offset by the same difference MOVE L CENTER 1 ; Center on the block OPEN 5 ; Open the hand CHANGE NIL,0,Z,90,0 ; and turn 90 deg. about z CENTER 1 ; re-center to grasp the block MOVE M ; and move away to M

No.	17	18
名　称	VML 〔Virtual Machine Language〕	PASLA 〔Programmable Assembly robot Language〕
開発元 （国）	LADSEB-CNR （イタリア）	日本電気(株) （日　本）
出　典	Proceedings 11th ISIR, October 1981	ロボット，No.32，October 1981
概 要	小規模な自動組立てのための，コンパクトで有効なソフトウェア・システムである．マイクロネットワークのためのコマンド言語でもあり，64KB以上のメモリーをもつ，16bitマイコン上でランし，PASCALで記述されている．オペレータおよびマニピュレータからの緊急ブレイク機能がある．	インタープリータ型式であるが，中間コードに変更することにより，実行速度が多少早くなっている． 座標教示機能，シーケンサ機能などがあり，また，特殊なOSを使用しないで，インストラクションを逐次実行する方式を採用している． 動作指示言語であり，基本インストラクションは，20ある． NEBOTの制御に使用されている．使用されているCPUは，8080であり，メモリーは，ROM24K, RAM24Kである．

記
述
例
	PROCESS;
	SCL_VAR(TEMPOS); LABEL(LAB1,LAB2);
	BEGIN;
	PARK_HARM;
	{we realalized the right arm}.
	PUSH_CONST(C1); GET_VAL(ARRIND);
	{we made the assignment arrind::=1}
	SEND(INIT);
	{we released the message init}
	LAB1:REGION(ARRIND);
	{we have reserved the variable arrind for updating his value}
	PUT_VAL(ARRIND); PUSH_CONST(C10); COMPARE;
	{we test if arrind is equal to 10}
	JUMPC(LAB2);
	{the process will stop if all positions of pos have been exhausted}
	GET_VAL (TEMPOS); PUSH_CONST(C1); PLUS; GET_VAL(ARRIND);
	{we made the assignments tempos::=arrind and arrind::=arrind+1}
	END_REGION(ARRIND);
	{we released the variable arrind}
	PUSH_CONST(R); MOVE_RPNT; OPERATE(DEVICE1);
	{we moved the right arm to the point R and operated device1}
	PUT_VAL(TEMPOS); IND_PUT_ARVAL(POS); MOVE_RPNT; OPERATE(DEVICE2)
	{we moved the right arm to the point pos [tempos] and
	we operated device2}
	JUMP(LAB1);
	LAB2:END_REGION(ARRIND);
	{we released the variable arrind}
	END;
	PROCESS;
	SCL_VAR(TEMPOS); LABEL(LAB1,LAB2);
	BEGIN;
	PARK_LARM; RECEIVE(INIT);
	{we block the process till the message init has been received}
	LAB1:REGION(ARRIND)
	PUT_VAL(ARRIND); PUSH_CONST(C10); COMPARE; JUMPC(LAB2);
	GET_VAL(TEMPOS); PUSH_CONST(C1); PLUS; GET_VAL(ARRIND);
	END_REGION(ARRIND);
	PUSH CONST(Q); MOVE_LPNT; OPERATE(DEVICE3);
		FILE−NO−001
		
		010 MZ　0100.00
		020 DF　001,001
		030 DF　002,002
		040 DF　003,005
		050 XY　M001
		060 CF　002
		070 XY　M002
		080 OF　001
		090 DL　050
		100 DC　003
		110 JX　003,050
		120 XY　M001
		130 MB
		
		FILE−NO−002
		
		010 MZ　0150.00
		020 OX　001
		030 DL　050
		040 MZ　0100.00
		050 RE

No.	19	20
名　称	PLAW 〔Programming Language for Welding robot〕	SERF 〔Sankyo Easy Robotic Formula〕
開発元 （国）	（株）小松製作所 （日　本）	（株）三協精機製作所 （日　本）
出　典	オートメーション，1982/4，Vol.27 No.4	オートメーション，1982/4，Vol.27 No.4
概 要	溶接ロボットのためのプログラミング言語であり，以下の特徴がある． ①各種センサーからの情報による任意の適応制御システムを実現する． ②溶接作業そのものに適した命令文をもつ． ③ティーチング・ポイントに対しての相対位置移動機能を有している． マイコンで制御でき，より少ないステップのプログラミングと，より少ないティーチ・ポイントが可能である．	主に組立用をねらって設計され，部品の挿入，圧入，箱詰，ハンダ付けなどに応用されるSKILAMの制御用言語である．CPUに6800を使用している． 言語は，簡単で理解しやすく学習に時間を要さないことを目標にしている．そのため，全体としてマニピュレータの動作命令に，FORTRANやBASICなどに似た言語体系を組み合わせたものになっている．
記 述 例	1 WDATA 200, 150, 400, 350 2 SPEED 0 MOVE B10 SPEED 120 MOVE P10 SENSE 1,108 MOVE P10 CHECK E20,5 SENSE 1,140 MOVE E20(0,0,-16)A CHECK E21,5 MOVE E21(0,-16,0)C TRANS P11 SPEED 80 ADATA 30,35 WEAV 30,50 SENSE 158,2 CALL 20 MEMO SPEED 0 WDATA 200, 155, 400, 355 MOVE P14 MOVE P11 WEAV 30,40 CALL 20 PLAY : : WAIT 8 GOTO 2 20 SARC CMOVE P11,P12,P13 SENSE CRATR 25,29,20 RET ON 12 WAIT 10 OFF 12 GOTO 2 END	NAME LU3-TEST TYPE L REM POINT ASIGN P(1) : X1,Y1,AS1 P(2) : 0,300,0 P(3) : 400,300,90 REM MATRIX ASIGN MX(1) : X(-600,20,4),Y(2,-22,4) REM FIXED SEAQUENCE ASIGN S(1) : L1(ON),I1(ON),L2(ON),DLY(0.5,L1(OF),I2(ON) S(2) : L1(ON),I1(ON),L2(OF),DLY(0.5),L1(OF),I2(ON) REM ACTION DIA = 50 DIR = 2*3.1416/16 MV / HOME,TR(12,0) DO K = 0 TO 16 STEP 1 CALL 100 MV / MX(1),S(1),P(2),S(2) 10 IF I3 = OFF GO 10 MV / S(1),LIN(10),P(3),S(2) 20 IF I4 = OFF GO 20 MV / S(1),LIN(0),P(1),S(2) NEXT K GO 200 100 X1 = -300+DIA/2*COS(DIR*K) Y1 = 300+DIA/2*SIN(DIR*K) AS1 = DIR*K RETURN 200 END

No.	21	22
名　称	高水準ロボット言語	ロボット工業会言語
開発元 (国)	東京大学　機械工学科 (日　本)	(社)日本産業用ロボット工業会 (日　本)
出　典	Proceedings 11th ISIR, October 1981	コンピュータ・アシステッド・ロボットシステム・エンジニアリングのシステムデザイン調査報告書　昭和56年3月
概 要	機械の組立，修理，分解のためのマニピュレータ動作を記述するための言語であり，ALを基礎としている．マクロによる言語の拡張と環境モデル・マネジャーに特色がある． 実験用に作成されたユニバーサル・アームを用いてインプリメントされた． 計画時システムは，コンパイラと環境モデル・マネジャーから成り，MELCOM COSMO 700 TSS上のLISPで作られている．実行時システムは，軌道計算器，アーム・コントローラ，アームから成り，軌道計算器は，ECLIPSE S/140に，Paulの軌道計算法を用い，FORTRANで書かれている．	作業指示言語であり，その役割は，次のような点にある． ①ロボットの制御プログラム作成を容易にする． ②プランニング時点で軌道計算を行なうことにより，現場でのティーチングを削減する． ③作業のプランニング全体を見やすく整理する． すなわち，利用者向けのロボット言語である． また，組立作業をロボットの操作とは独立に，部品間の組立関係という言葉で定義しようというObject Levelのアプローチをとっている． ALに似た言語体系である．
記 述 例	(PROGRAM EX2 (DECLARE INITIAL-POINT COORDINATES) (DECLARE FINAL-POINT COORDINATES) (DECLARE BOX PART) (DECLARE BOX-GRASP COORDINATES) (ASSERT INITIAL-POINT LOCATE-AT (COORDINATES NILROT (VECTOR 300 300 100))) (ASSERT FINAL-POINT LOCATE-AT (COORDINATES NILROT (VECTOR 300 -300 100))) (AFFIX BOX INITIAL-POINT NILTRANS NONRIGIDLY) (AFFIX BOX-GRASP BOX (TRANS (ROTATION YHAT 180) (VECTOR 10 10 5)) RIGIDLY) (OPEN ARM 50) (MOVE ARM BOX-GRASP (APPROACH (VECTOR 0 0 -10))) (CLOSE ARM 20) (MOVE BOX FINAL-POINT (DEPARTURE (VECTOR 0 0 10)) (APPROACH (VECTOR 0 0 10))) (OPEN ARM 50) (MOVE ARM PARK (DEPARTURE (VECTOR 0 0 -10)))))	PROCESURE DEFINITION; PROC main, start : PARK hand1 ; (hand1 をpark-positionへ移動する。) OPEN hand1 TO MAX ; (ハンドを全開する。) GRASP part-A ; (park-position (移動空間から，部品供給SUB-SPACEにENTRANCE出口を通過して入り，approach点に結合せする。つかむために必要な形合せを行い，part-Aに接近し，つかみ，そのまま departure点までさがる。) PLACE part-A ON assy-station ; (ENTRANCE出口を通過し，移動空間に出て，組立作業SUB-SPACE へ，そのENTRANCE出口を通って入る。assy-stationのapproach点にpart-Aを位置づけ，結合せする。置くために必要な形合せを行い，assy-stationのTSPとpart-AのESPが一致する点まで接近する(part-Aを置く)。) RELEASE part-A ; (ハンドを全開し，departure点までさがる。) CLAMP part-A ; (万力でpart-Aを固定する。) GRASP part-B ; (組立作業SUB-SPACEのENTRANCE出口を通って，移動空間に出て，部品供給SUB-SPACEに向と同時に入り，part-Bのapproach点に結合せする。形合せを行い，part-Bに接近し，つかみ，そのまま departure 点までさがる。) FIT part-B TO part-A WITH FORCE(ZHAT)=200*GM ; (移動空間に出て，組立作業SUB-SPACEへ入り，part-Aのapproach 点にpart-Bを結合せし，形合せする。part-AのTSPとpart-BのESPが一致する点まで接近する。そして，挿入れを行う。 (荷重値圧(押し込み力)200g(一定)，はめ合い時間5秒以内) IF DURATION>5*SEC ; THEN ; CALL reject1 ; GO TO start ; END ; (5秒たってもはめ合いが完了しない場合，3部品ともreject1-boxへ捨て，はじめに戻る。)