展 望

ロボット研究開発プラットフォームにおけるソフトウェア

Software Platform for Robotics Research

小笠原

司* *電子技術総合研究所

Tsukasa Ogasawara* *Electrotechnical Laboratory

1. はじめに

ロボット研究はシステム構築を目指したものであり、その中のソフトウェアの占める比重は大きい。従来のロボット研究を見ていると要素技術の研究を行い、それを使えるようにするまでが精一杯であり、確立された要素技術をツールとして使い込み、さらなる発展を試みるに至らないという過程を繰り返すことが多かった。その理由の一つには研究開発プラットフォームが存在しなかったことが大きい。特に、ソフトウェア面でのプラットフォームは数える程しか存在しない。このままではより複雑化するロボットシステムの構成技術は一向に確立されず、今後のロボット研究展開上大きな障害となりうる。幸い、この様な問題意識を持つ研究者は多く、今回の研究開発プラットフォーム特集の発行もそれに対する挑戦の始まりとでも言うことができる。本稿では、研究開発プラットフォームにおけるソフトウェアの機能について考察するとともに、今後のロボットソフトウェアの研究課題について展望する。

2. 研究開発プラットフォームとソフトウェアの現状

筆者が大学院に進学しロボットのソフトウェアシステムの研究を始めた1970年代末は大型計算機全盛の時代であった。300/1200[bps]のモデムを介して端末からログインしロボット言語処理系の開発を行っていた。それが、博士論文を書くころには16ビットミニコン上にもコンパクトながら使い勝手のよいOSが誕生し、それを用いてプログラミングシステム[1]を構築することができた。当時、M 68000 相当のCPU、1[MB]の主記憶、10[MB]程度のハードディスクを備えた持ち運べるマシンがあれば、それにLisp処理系を乗せてロボットの研究には十分使えるのになどということを研究室で話していた記憶がある。それ

は5年もしないうちに実現されてしまった。当時羨望の的であった32ビットスーパーミニコンは今や過去の遺物と化してしまった。90年代に入りワークステーションが全盛となり、処理速度の画期的向上、GUI技術の進歩、周辺デバイスの進歩は想像をはるかに上回るスピードで起こった。ロボットのプログラミング環境もそれらの技術を用いた高度なものが開発され現在に至っている。さらに最近のパーソナルコンピュータの小型・高機能化、低価格化により、誰でも同じ様な環境でロボット研究開発が行える基盤が整ってきた。

以下にソフトウェアプラットフォームの要素機能を中心に研究の流れを振り返ってみる。

2.1 ロボット言語

ロボット研究開発プラットフォームとしては作業を行うマニピュレータのためのものが多い。ロボット研究がマニピュレータ研究から本格的に始まったのも一因であるが、生産現場への応用がまず試みられた分野であるからであろう。

マニピュレータとしてはStanfordアーム、PUMAなどが事実上の標準として用いられ、これらを動作させるためのプログラミングの研究が1970年代、80年代に精力的に行われた。いわゆるロボット言語の研究である。作業記述方式の研究が中心で動作レベルの言語から対象物レベル、作業目標レベルといった分類がなされ、AL[2]、VAL[3]、AML[4]などの動作レベル言語、RAPT[5]、AUTOPASS[6]、LAMA[7]などの対象物レベルの言語が研究開発された。

言語の機能を向上する上で作業環境を計算機内部で表現するためのモデリングの機能は重要であり、RAPT、AUTOPASS、LAMAなどではこれらの機能についての議論がなされた。しかし、研究室レベルの域を出るものではなかった。モデリング機能自体がまだ不十分であり、複雑な環境を表現できなかったのも一因である。

VALはPUMAの制御用、AMLはIBM製のマニピュレータ制御用として広く使われ、ある意味でのプラットフォーム的役割を果たした。

原稿受付 1995年12月4日

キーワード: Robot System, Programming System, Software System

*〒305 茨城県つくば市梅園1-1-4

*Umezono, Tsukuba-shi, Ibaraki

ロボット言語としては、多種類のロボットの作業記述を行えることが一つの目標となる。それを目指し標準化の研究も行われており、国内ではロボット言語 SLIM、中間コード STROLIC が JIS で規格化されている[8]。

2.2 実時間 OS

ロボットシステムを動作させる上で、OS レベルの機能も重要である。

研究用に一般的に用いられている Unix 上で C 言語を用いたロボットのプログラミングを可能にしたのが RCCL [9]である。標準 UNIX では実時間性に劣るため、Real Time Unix が開発され、UNIX のデバイスドライバを用いてメカトロ機器の I/O を標準化する試みもある[10]。標準化によりデバイス依存部分を隠蔽し、ソフトウェア開発の効率化を図っている。

商用の実時間 OS を用い、その上に C、C++を用いて制御ソフトウェアを書いているのが一般的なロボット研究の方式と言えるであろう。VxWorks、Lynx など Unix との親和性を備えたものが多く用いられている。しかし、コンパクトな自律移動ロボットとなると組み込み可能な計算機のサイズ、使用するインターフェースの多さから、自前の OS を開発する例も多く見られる。COL[11]や山彦の RCS[12]などがその例である。

マニピュレータに関してもシステムの拡張性を意識し、モジュラリティを追及した CMU の CHIMERA[13]などのロボット用 OS の研究がある。

ネットワークを介して複数台の異機種計算機が接続された環境でもそれらを統合したシステムの並列・分散処理機能を実現し、容易に利用可能にするための分散 OS の研究も行われている。OZ+[14]はそのような OS の一つである。

2.3 ロボットの設計支援

ロボットシステムを開発するに当たっては、その運動学、動力学の解析、および、メカニズムを制御するための制御系の設計が大きな問題となる。これらを支援する設計システムの研究もソフトウェアプラットフォームとしては重要である。

運動学の解析に関しては、数式処理システムを用い、ロボット解析用のライブラリを開発して用いている。Robotica[15]は Mathematica を、ROSAM[16]は MAPLE をベースとしたロボット解析システムとして精力的に研究開発が行われており、ともに本特集号に付属の CD-ROM にも収められているソフトウェアである。

制御系の設計に関しては MATLAB を用いる例も多い。ブロックダイアグラムで制御系を設計し、その応答のシミュレーションを経て、ターゲットのソースコードを生成することができる。さらに、さまざまなライブラリも備わっており、一般的な制御系設計システムとして広く用い

られている。MATLAB によるロボット解析ツールも上記 CD-ROM に集録されている。

2.4 プログラミングシステム

ロボット研究を進めていく上で、上記に述べたようなソフトウェアコンポーネントを統合していくソフトウェアプラットフォームが必要となる。ロボットは総合システムであり、システムを用いて確立された新たな技術は逐次システムに組み込み、その機能を充実させていかなければならない。ある場合にはハードウェアの大幅な追加変更もありうる。それにも耐えうるだけのモジュラリティ、拡張性がソフトウェアプラットフォームには要求される。

いくつかの研究機関ではプログラミング環境の開発が行われており、そのプラットフォームの上に各研究者の成果が逐次統合されている。この手法なくして、今後の大規模なロボットシステムの構築はなし得ない。

このようなプログラミングシステムの一つに EusLisp [17]がある。Common Lisp 並のプログラミング言語を対象指向プログラミング機能を用いて実現したものである。さらに、ロボット研究用に幾何モデリング機能、幾何学演算機能、ネットワーク通信機能が装備されている。最近では、マルチスレッドによる並列処理機能が追加された。本特集号付属の CD-ROM にも集録されている。

EusLisp は上位のプログラミングレベルでのプラットフォームを目指したものである。そのため、制御レベルでの細かな I/O、高速実時間性、などについては完全には対応できない。ランタイムのプラットフォームを別途用意し、ネットワークを介した通信機能によりこれらを統合する方式にならざるを得ない。ランタイムのソフトウェアプラットフォームとしては、前述の CHIMERA などの実時間 OS に基づいたものがある。

自律移動ロボットに関してもプログラミングシステムの開発が行われている[18]。また、最近ではエージェントの概念に基づくプログラミングシステムも研究されている[19]。

3. 今後の展望

ロボット研究開発プラットフォームにおけるソフトウェアは以上のような状況である。プラットフォームの要素機能が完全にはクリアされておらず、まだまだ統一的なソフトウェアプラットフォームという状況にはないが、それに近づいたものが生まれつつあるという情勢である。では、今後の解決すべき問題点は何であろうか？ 以下ではその点に関して展望を行う。

3.1 ロボット用 OS

ロボットの特徴として、多数のセンサ群を高速度で並列に処理して対象を認識し、状況に応じた適応行動を行うという点が上げられる。これらの処理は非同期処理であり、

イベント駆動方式で行われなければならない。

これを実現する上で、基本ソフトウェアのさらなる機能向上が要求される。高度の並列処理のサポート、センサの付加の容易性、ハードウェアやCPUに依存しないマルチベンダ、マルチプラットフォームの対応などである。特に超並列・超分散機能は従来のロボット制御用に用いられているOSには欠けている点であり、十分な研究が必要である。また、実時間性を有するネットワーク機能、ネットワーク媒体に依存しない接続性も要求される。並列計算システム上で動作する実時間ロボットOSの開発が待たれる。

3.2 ユーザインターフェース

ロボットを使いこなす上でユーザインターフェースは重要な課題である。従来のロボット言語、教示システムの研究も、結局は使いやすさを追及する一つのアプローチであった。仮想現実感技術の向上は、この分野に画期的な革新をもたらす可能性がある[20]。計算機内部での動作シミュレーションを行い、グラフィクス技術を用いた仮想環境による作業教示技術に関しては、いくつかの研究が始まっている。初期の研究としてマルチメディアディスプレイを用いた作業教示法[21]があげられる。

動作指示を行うための指示デバイスとしては力帰還型のマスターアームを用いた方式が一般的である。今後、新たなデバイスが開発された場合には、容易にこれを導入できるだけの拡張性をソフトウェアプラットフォームは有していなければならない。

GUI機能の整備も必要となる。ウィンドウベースのGUI機能と入力デバイスの統合、そして、内部情報の可視化機能の組み込みと課題は多い。この分野は技術革新が激しく、ソフトウェアプラットフォームは新たな技術を逐次統合できる機能が要求される。

3.3 モデリング

幾何モデルをベースとした環境モデルに基づく作業実行方式は、従来より知能ロボット技術として確立されてきた。しかし、多面体モデルに基づくため、モデル化の誤差が大きいため、実環境に十分には対応できないものであった。これを打開する新たな制御アーキテクチャとしてビヘービアベースト制御方式が提唱された。これらを融合した形式の制御を実現するための機能をテストベッドに組み込まなければならない。また、自由曲面、さらに柔軟物まで扱えるような幾何モデルの拡張、環境モデルを構成する機能の組み込み、センサからのリアルタイムフィードバックによりモデルを更新する技術など、これらを実現するためのプログラミングパラダイムの導入が必要となる。ソフトウェアプラットフォームにはさまざまなアーキテクチャの概念の試作のためのプログラミング機能が要求される。

3.4 モジュールティ

ロボットの研究対象が広がるにつれ、扱うロボットの種

類は格段に増えてきた。従来は1台のマニピュレータのプログラミング機能を整備すればよかったが、現在では、マニピュレータの種類、台数、また、作業によってはマニピュレータと自律移動ロボットを操るという具合に、多種多様な形態が考えられる。これに対応するためにはテストベッドがソフトウェアのモジュールティを有していなければならない。ロボットのソフトウェアシステムを構成することがあたかも部品を組み立てて製品を作るかのごとく達成できるのが理想である[22]。そのソフトウェア部品は対象となるロボット、要素機器、さらには、基本ソフトウェアごとに整備されており、研究者はデータベースを検索して自分の研究に必要なモジュールを容易に集めることができるという具合である。

3.5 ネットワーク機能

昨今のネットワーク技術の著しい進歩はロボット研究にも大きな影響を及ぼすものである。従来の単体ロボットの研究に止まらず、複数台ロボット群の協調制御、空間に分布したロボット要素を統合する手法、広域ネットワークを用いた遠隔ロボティクスなどの研究が必要となる[23]。その際にも、そのベースとなるソフトウェアシステムが重要な位置を占める。超遠距離通信における時間遅れ、通信量の問題、実時間性を有するネットワークソフトウェアの機能実現など、通信関連の技術を支援するソフトウェア環境が重要となる。ネットワークは狭義の広域ネットワークによる接続ばかりではなく、ロボット内部のCPUバスを介した接続も広義のネットワークである。このような内部バスによる密な結合から遠距離の疎な結合まで各種のネットワークを用いたあらゆるレベルでの結合形態が考えられる。これらを統合的に扱うことのできる通信メカニズムが必要とされる。また、そのようなネットワークに接続する要素機器に関してはインターフェースの標準化が重要である。そのためのソフトウェアのモジュール化の必要性は前述のとおりである。

3.6 統合化プラットフォーム

従来のロボットの統合システムを実現したソフトウェアは人工知能寄りのアプローチを取っていた。推論、認識、計画機能などを人工知能向き言語で統合するという手法である。一方、制御レベルでは、実時間応答性は必須であり、その機能を有する実時間OSを核とするテストベッドを構築していた。また、設計の分野では数式処理システムを用いたアプローチが見られる。現状ではここまでであるが、製造分野における動向から類推されるように、さらなる統合化が今後の課題となる。製造業における発注から生産までの統合をめざすCALSのようなソフトウェアの考え方がロボット工学においても必要とされる。設計からプログラミング、コントロール、そして、良好なユーザインターフェースまでをネットワーク技術により統合一体化するこ

とにより、統合化研究開発プラットフォームとしてロボット研究に役立つプログラミング環境の開発が期待される。さらに、人工生命、創発システムなどの新たな概念を積極的に取り込める拡張性は常に要求される。

その一方、統合化することによりシステムが複雑になってしまい、全体の構成がよく見えなくなってしまうという弊害も考えられる。制御中心の研究を行うのによけいなモジュールまで含んだ重いシステムを覚えなければならないのは厄介である。カーネル部を中心に必要最小限の機能を抽出したランタイムシステムを柔軟に構成できる機能が必要であろう。最近の事務用アプリケーションソフトウェアのアドオン機能が参考になる。ロボットソフトウェアを作る上でも十分配慮しなければならない。

4. お わ り に

ロボットは動かしてこそ価値のあるものである。それを行うためには共通のソフトウェアプラットフォームを多数の研究者が使用し、各々の研究成果をそのプラットフォームに蓄積し、それらを交換しながらシステムを開発する過程が必要である。最近のロボット研究にはシミュレーションで終わってしまうものが少なからず見受けられる。決められた期間で成果を出すためには、ある程度仕方のないことかもしれない。ロボットのシステムが複雑、大規模化しているため、個人の能力では限られたことしかできなくなっているのも事実である。いまこそソフトウェアプラットフォームを真剣に議論しないといけない時期に来ている。それにより、ロボット工学の新たな展開を期待するものである。

参 考 文 献

- [1] 小笠原, 井上: “知能ロボットプログラミングシステム COSMOS”, 日本ロボット学会誌, vol. 2, no. 6, pp. 507-525, 1984.
- [2] M. Mujtaba and R. Goldman: AL User's Manual, Stanford AI Lab., 1979.
- [3] Unimation Inc.: User's Guide to VAL, Unimation Inc., 1980.
- [4] R. Taylor, P. Summers and J. Meyer: “AML: A Manufacturing Language,” International Journal of Robotics Research, vol. 1, no. 3, pp. 19-41, 1982.
- [5] R. Popplestone, A. Ambler and I. Bellos: “RAPT: A Language for Describing Assemblies,” Industrial Robot, vol. 5, no. 3, pp. 131-137, 1978.
- [6] L. Lieberman and M. Wesley: “AUTOPASS: An Automatic Programming System for Computer Controlled Mechanical Assembly,” IBM Journal of Research and Development, vol. 21, no. 4, pp. 321-333, 1977.
- [7] T. Lozano-Pérez and P. Winston: “LAMA: A Language for Automatic Mechanical Assembly,” Proc. of 5th IJCAI, pp. 710-716, 1977.
- [8] 松元, 安藤: “ロボット言語 SLIM のためのプログラミング環境”, ロボティクス・メカトロニクス講演会 '92 講演論文集, vol. A, pp. 501-504, 1992.
- [9] V. Hayward and R. Paul: “Robot Manipulator Control under UNIX RCCL: A Robot Control 'C' Library,” International Journal of Robotics Research, vol. 5, no. 4, pp. 94-111, 1986.
- [10] E. Nicolson: “Standardizing I/O for Mechatronic Systems (SIOMS) using Real Time UNIX Device Drivers,” 1994 IEEE International Conference on Robotics and Automation, pp. 3489-3494, 1994.
- [11] 光石, 下山, 三浦: “並行プロセス指向言語 COL の開発”, 日本ロボット学会誌, vol. 3, no. 6, pp. 509-520, 1985.
- [12] 津田, 金山, 油田: “知能ロボット用 OS-RCS のインプリメンテーション”, 日本ロボット学会誌, vol. 3, no. 5, pp. 432-443, 1985.
- [13] D. Stewart, E. Schmitz and P. Khosla: “Implementing Real-Time Robotic Systems Using CHIMERA II,” IEEE International Conference on Robotics and Automation, pp. 598-603, 1990.
- [14] 塚本, 浜崎: “オブジェクト指向開放型分散システム: OZ+”, 情報処理, vol. 36, no. 8, pp. 715-720, 1995.
- [15] J. Nethery and M. Spong: “Robotica: A Mathematica Package for Robot Analysis,” IEEE Robotics and Automation Magazine, vol. 1, no. 1, pp. 13-20, 1994.
- [16] 川崎, 島田: “数式処理 MAPLE によるロボット解析システム”, 第 12 回日本ロボット学会学術講演会予稿集, pp. 817-818, 1994.
- [17] 松井, 関口: “マルチスレッドを用いた並列 EusLisp の設計と実現”, 情報処理学会論文誌, vol. 36, no. 8, pp. 1885-1896, 1995.
- [18] 鈴木, 木元, 油田: “移動ロボットの自律行動のためのプログラム開発環境の構築”, 日本ロボット学会誌, vol. 12, no. 3, pp. 163-172, 1994.
- [19] F. Zanichelli, et al.: “A Multi-Agent Framework and Programming Environment for Autonomous Robotics,” 1994 IEEE International Conference on Robotics and Automation, pp. 3501-3507, 1994.
- [20] 藤本: 人工現実感の展開, コロナ社, 1994.
- [21] 松井, 塚本: “マルチメディアディスプレイを用いた統合型遠隔ロボット制御法”, 日本ロボット学会誌, vol. 6, no. 4, pp. 33-41, 1988.
- [22] H. Tsukune, et al.: “Modular Manufacturing,” Journal of Intelligent Manufacturing, vol. 4, no. 2, pp. 163-181, 1993.
- [23] 小笠原, 他: “ユービキタスオートノミー——ロボット遍在技術の確立をめざして—”, 第 13 回日本ロボット学会学術講演会予稿集, pp. 1197-1198, 1995.

小笠原司 (Tsukasa Ogasawara)

1955 年 10 月 16 日生。1978 年東京大学工学部計数工学科卒業, 83 年同大学院情報工学専門課程博士課程修了。同年通産省工業技術院電子技術総合研究所入所。現在, 知能システム部行動知能研究室室長。1993 年-1994 年, ドイツ, カールスルーエ大学客員研究員。知能ロボットの研究に従事。工学博士。計測自動制御学会, 情報処理学会などの会員。(日本ロボット学会正会員)