

解説



ソフトウェア・ツールとその位置付け†

國井 利 泰††

1. 手動ツールから自動ツールへ

ソフトウェア・ツール（ソフトウェア工具）というと、常識的にはソフトウェア作製の手助けをする便利な工具としての、ある種のソフトウェア群を指すものと解されている。特にその中でも、数年前まではプログラム作製時に極めて便利なテキスト・エディタなどのプログラミング・ツールの類を指すのが通例であった¹⁾。しかし、最近情報産業の急激な発展に伴って、ソフトウェアもかつての単なるコードの集まりあるいはプログラムから始まって、内部・外部仕様書の整備された通常のソフトウェアを経て、機種に拘らず使えてオンライン・ドキュメンテーションも完備し、保守性も極めて良いソフトウェア製品へと次第に重点が移って来るに至った²⁾。それと呼応し、それを支援するものとしてのソフトウェア・ツールも、その重点がプログラミング・ツール、通常ソフトウェア生産用ツールから、現在はソフトウェア製品生産用自動ツール³⁾に次第に移って来た。

これをもうちょっと一般になじみの深い家具生産に例えれば、かつては手作りで楽しくひなたぼっこしながら木をナイフなどの非自動道具（ツール）類で削って一つ一つ作られていた机・いすなどが、今や数値制御プレス機などの自動工具類の並んだ工場で流れ作業で作られて現代社会の家庭ならびにオフィスにおける大量供給、低価格の需要に応えるようになったようなものである。趣味的、価格的には、最高の道楽はいぜんとして手作りのものである。それはいわば、製造過程の不確かさとか偶然性から生まれる創造性を楽しむ芸術の領域である。オフィスや家庭で数少ない高価な芸術作品の創造やあるいは使用は、場面的にも、价格的にも極めて限定されざるを得ない。同時に、工業芸術の発達により、大量工業生産に別の型の芸術一例え

ば工業デザインなどが導入され、大きな発展を遂げてきているのも周知の通りである。

家具とソフトウェアの違いは、ソフトウェアの場合は、あるいは従業員全体の給与を計算し、あるいは会社全体の経理を司り、あるいは列車の運行を実行・監視し、あるいは原子力発電の制御を行うなど、いわば社会の中核的機能の実行を担当していることが少なくなく、したがって多くの場合極めて高い信頼性が要求される。そして、ソフトウェア・ツール群も、それによって作られたもの（ソフトウェア）は正確にあらかじめ規定された通り動く可能性のある自動ツール群の方が信頼性向上の面からも好ましいことになる。いうまでもないことであるが、いったんこれらの自動ツール群が正しく作られていることがテストや繰返し使用の結果わかれば一もちろんこれ自体大きな仕事であるが一、自動ツールは入力さえ誤り無く与えれば、誤りのない結果を与えてくれる。それに対して、非自動ツールの場合は、入力にもツールにも誤りが無くても、結果は誤り得る。なぜなら使い方を誤る可能性があるから。したがって、単に省力化のためだけでなく、生産されるソフトの信頼性向上のためにも、可能な限り自動化すべきであるという論拠はここにもある。

2. 自動ツールの目標と環境

コンピュータはもとより、ミシンなどの家電製品から工作機械、航空機及び列車運行予約システムなどの社会システムに至るまでほとんどの工業製品に占めるソフトウェアの割合が増大している。ソフトウェア製品生産の道具としてのソフトウェア工具の社会的・産業的役割も当然増大の一途をたどっている。このような状況で、自動ツールを使いやすい環境の整備がますます緊急となっている。もっと詳しく検討してみるとあらゆる道具あるいは工具に共通のことであるが、ソフトウェア工具の場合にも、その有効性は以下の3項目に大きく依存する：

- 構成（アーキテクチャ）：道具の出来の良し悪し
- 使用目標：どのような適用分野あるいは目標をも

† Software Tools and Their Roles by Toshiyasu L. KUNII
(Department of Information Science, Faculty of Science,
The University of Tokyo).

†† 東京大学理学部情報科学科

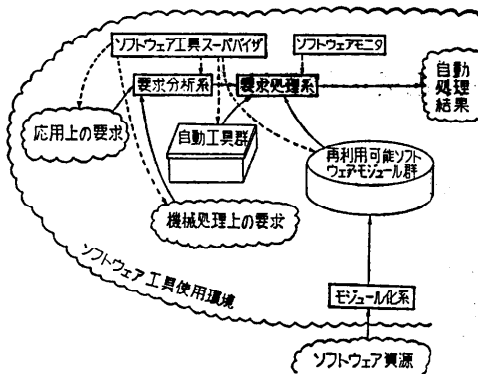


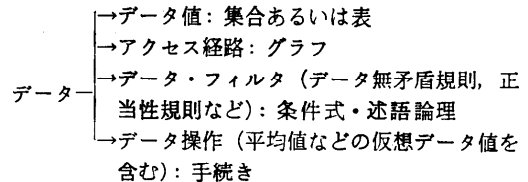
図-1 自動ソフトウェア工具使用環境

って使用するか

- 使用環境：使われる場、並びに状況設定がどのようなものであるか、またどの位整備されているか（道具を使う側の体制整備などを含む）

工具のアーキテクチャは使用目標と環境に合致するように立てられる。使用目標としては、通説によると80%を越したといわれているソフトウェアのハードウェアに対して占める比重の増加に対する工業的取組としての「ソフトウェア製品生産の自動化」、ソフトウェア保守コストの際限のない増大への対策としての「保守コスト軽減」などがあげられる。このような目標に対応して、生産用ソフトウェア工具のアーキテクチャの重点も手動工具から自動工具³⁾へ移行し、保守用工具も問題が起ってから使う後手保守用工具から製品の信頼度自体を高めて保守の必要性を削減する高

信頼度設計・実現用工具⁴⁾に移行しつつある。同じく使用環境も、このような使用目標を可能にするように整備されつつある。例えば、自動工具群を応用系あるいは処理機械系の要求に応じて呼出すソフト工具スーパーバイザ、これ等の要求を機械による自動処理が可能な形式に記述するための要求仕様形式化技術⁵⁾、ソフトウェア資源をこれ等の自動工具により繰返し利用可能な形—これを再利用可能ソフトウェア・モジュール(Reusable Software Modules)と呼ぶことにしよう—に分解してくれるモジュール化系などである(図-1参照)。コンピュータ・システムの処理能力を測定するのに用いられるソフトウェア・モニタ⁶⁾もかなりすぐれたものが市販されている⁷⁾。個々の機種あるいは応用に合わせて生産されたソフトウェアを、機種・応用の変化あるいは交代があった場合に自動的に新環境に再適応させるシステム構成を進化型アーキテクチャ(Evolutionary Architecture)と呼ぶ¹²⁾。その設計用グラフィックス・データベースへの応用を例として、例えばソフト資源の重要要素であるデータをいかに再利用可能にモジュール化するかを考察したのが文献13)である。データの場合だけとりあげても、一般にそれは以下ようになる：



この分解・統合のためのソフトウェア工具群と環境構

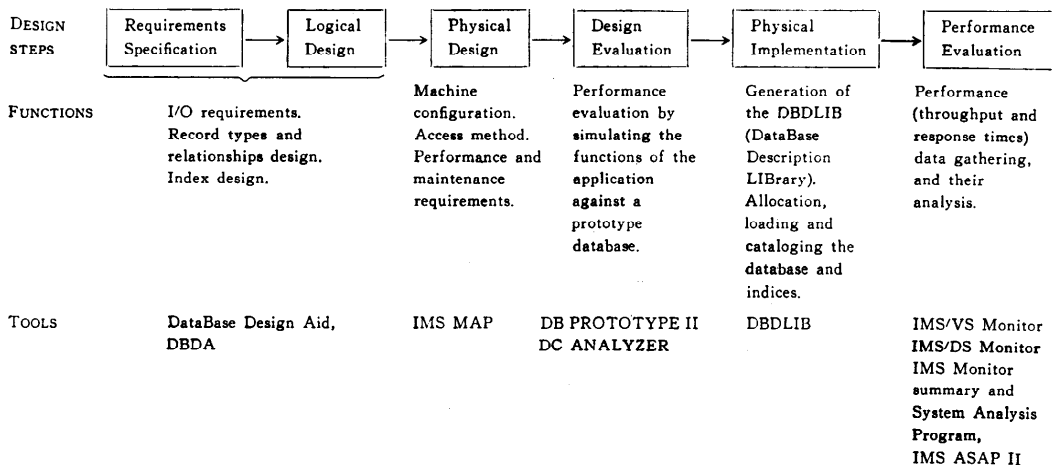


図-2 Tools for IMS Database Design

成は相当大規模になる。対象となるデータがこのように再利用可能な形にまで分解されていない場合には、その分解・統合は更に大仕掛けのものになり、当然、無駄が多くなる。IMS の場合を図-2 に示す。

古い歴史をもつソフトウェアの場合には、技術的に多くの制約を伴っていたので、やむを得ぬ事情があって分解の不十分なデータ・モジュールを、ソフトウェア工具群の大規模な開発により生かし続けている好例である。

3. 製品生涯サイクル管理におけるツールの位置

ソフトウェア製品生涯サイクル管理技術
ソフトウェア製品に対する社会的要求（需要）が発生し、計画が立てられ、設計・実現・テストを経て発売され、故障修理・保守・改善が行われる比較的長い時期を経て新製品に置き換えられるまでの、いわゆる製品生涯サイクル管理 (Products Life Cycle Management) を重点に置いた管理技術が最近重点的に考慮されるようになった⁸⁾。この技術は、自動設計・実現工具群のフル援用によりソフトウェア製品生産総コストを最小にする方法を総括的に包含する。

この技術のアプローチの概念的基礎となっているのは、応用上ある機械処理上の要求に応じて、計画段階で自動設計・実現工具群を駆使して設計・実現・テスト・保守が自動化できるように立案することにある。これは従来人海戦術により設計・実現・保守を行って

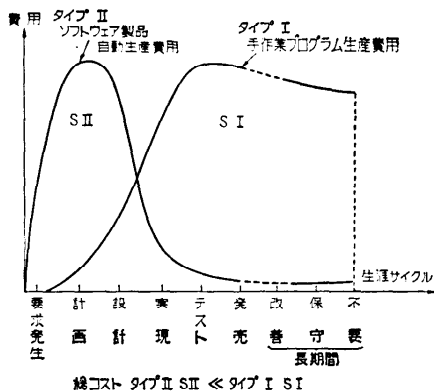
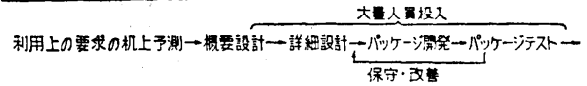


図-3 ソフトウェア製品生涯サイクルと生涯サイクル費用

例 工学科学データベースシステム開発

タイプ I: 人海戦術生産型



→ パッケージ結合試用 → 顧客実用
(応用のパッケージ呼出順序記述による)

タイプ II: 自動生産指向型

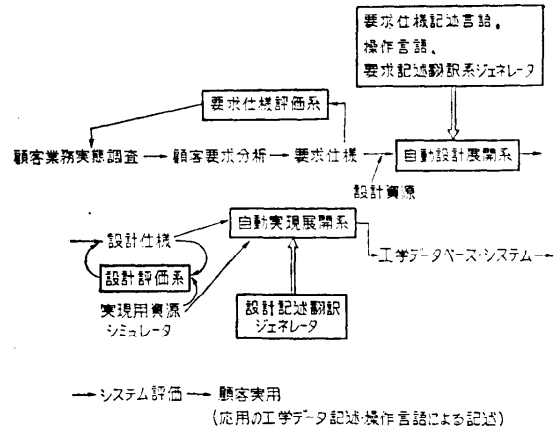


図-4 人海戦術型生産パターン (タイプ I) と自動生産指向パターン (タイプ II) の例

いた生産パターン (図-3 のタイプ I 曲線に示されるように) を新たに生産の初期段階である要求仕様・計画段階重点型 (図-3 のタイプ II 曲線参照) に変えるものである。それを可能にしているのが応用上あるいは機械処理上の要求などの記述のための要求仕様工具群 (記述言語) と自動工具群、ならびにソフトウェアモニターなどの環境支援工具群である。

タイプ II の典型として、宇宙船設計用等の米国宇宙航空局 (NASA) プロジェクト IPAD (Integrated Program for Aerovehicle Design) の工学・科学データベースを例にとって調べてみた^{9), 10)}。これまでのタイプ I の、人海戦術で多量にプログラムパッケージの集積を行う方式と異なり、設計・実現用ソフトウェア工具群を駆使してシステム生成を行う方式を提起し、実行していた (図-4 参照)。設計用データ記述・操作言語も LALR (1) パーサジェネレータを使用してほぼ自動的にコンパイラ生成を行い、処理能率を落とさずに満足 of いくまで度々言語仕様の改訂とそれによる機能向上を行える環境を整備していた。

この LALR (1) パーサジェネレータ¹¹⁾は、国防総省 DOD1 言語コンパイラ、カーネギメロン大学 ALPHARD 言語コンパイラ、TEKTRONIX 社知能端末用各種言語コンパイラなどの生成に広く使用実績があり、それ等の言語仕様の弾力的レベルアップを可能にしている。生成されたパーサは手でコーディングされたもの以上の実行速度を持ち、エラーメッセージも完備しているといわれ、典型的ソフトウェア生産用ソフトウェア工具の一つである。

以上の考察から明らかになった通り、ソフトウェア工具は、単に生産総コストの減少のみならず、ソフトウェア記述・操作言語を始めとするソフトウェアの飛躍的品質向上、納期短縮を可能にする。

参 考 文 献

- 1) Kernighan, B. W. and Plauger, P. J.: Software Tools, Addison Wesley, Reading, Massachusetts, 1976.
- 2) Gunther, R. C.: Management Methodology for Software Product Engineering, Wiley-Interscience, 1978.
- 3) Reifer, D. J.: Tools for Software Automation, ACM Professional Development Seminar, ACM National Conference, 1978.
- 4) Yau, S. S., Cheung, R. C. and Cochrane, D. C.: An Approach to Error-Resistant Software Design, Proc. of Second International Conference on Software Engineering, 429, Oct. 13-15, 1976, San Francisco.
- 5) Ramamoorthy, C. V. and Kim, K. H.: Software Monitors—Aiding Systematic Testing and Their Optimal Placement, Proc. 1st National Conference on Software Engineering, Sept., 1975 (IEEE Press, IEEE Catalogue No. 75CH 0992-8C).
- 6) 國井利泰監修「ソフトウェア工学—要求仕様技術—」bit 臨時増刊 1978 年 8 月号, 東京, 共立出版.
- 7) 例えば "LOOK", Applied Data Research, Route 206 Center CN-8, Princeton, New Jersey 08540.
- 8) 例えば Proceedings of the Second Software Life Cycle Management Workshop, August 21-22, 1978 (IEEE Press).
- 9) Browne, J. C.: Continuation Proposal for Research into Data Management Systems for Engineering/Scientific Computation, Department of Computer Sciences, The University of Texas at Austin, Austin, Texas 78712.
- 10) Elliott, L., Kunii, H. S. and Browne, J. C.: A Data Management System for Engineering and Scientific Computing, Proc. Conference on Scientific and Engineering Data Management, 197, May, 1978.
- 11) "I³ PG1: LALR (1) Parser Generator," Information International Inc., P.O. Box 8150, Austin, Texas 78712.
- 12) Kunii, T. L., Browne, J. C. and Kunii, H. S.: An Architecture for Evolutionary Database System Design, Proc. The IEEE Computer Society's Second International Computer Software & Applications Conference, Chicago, 382, Nov. 13-16, 1978.
- 13) Kunii, T. L. and Kunii, H. S.: Architecture of a Virtual Graphic Database System for Interactive Computer-Aided Design, Computer Aided Design Journal (in press).

(昭和 54 年 4 月 4 日受付)