

第10章 結 論

第10章 結 論

ソフトウェアの大規模化に伴い、その信頼性と生産性の向上はますます重要な課題となっている。そのため、1970年代には、まず構造化プログラミングをはじめとするプログラミング方法論の研究が盛んに行われ、理解容易なプログラム構造の重要性が認識された。そして、ソフトウェア開発技術の中で最も大きな役割を果たしてきたプログラミング言語についても、機械語への展開率を向上させる従来の高級言語に代わって、プログラミング方法論を反映させた言語の研究が行われてきた。その中で、言語設計に関する多くの提案がなされたが、言語処理系の開発にまで至ったものは少なく、実用に付されて効果をあげているものはさらに限られる。その主な理由は、新しいプログラミング方法論を具体化して実用化するまでにはなお多くの研究課題が残されているためである。

本論文は、このような背景の下に行ってきた、段階的詳細化とデータ抽象化を支援する言語SPLの開発と実用化に関する研究成果をまとめたものである。以下、各章ごとに得られた結果について述べる。

まず第1章の序論に続き、第2章で従来の諸研究との関係を明らかにした後、第3章においてSPLの設計思想について論じた。本研究で対象とする日立の制御用計算機HIDIC80用応用プログラムの記述には、従来、制御用Fortran (PCL)を用いていたが、主に2つの問題があった。その第1は、タスク間共通データへのアクセスエラーが多いことである。その原因分析に基づき、新言語では、アクセス権の厳密な規定、データ抽象化、および厳しい型チェックのための機能が必須であるという結論を得た。また、第2には、プログラムと設計書の対応が不明確で、保守が難しいという問題があった。これはプログラムの制御構造の複雑さによるものであることから、新言語では、段階的詳細化と構造化コーディングの機能が必須であるという結論を得た。そして、以上の結果に加えて、従来言語との親和性を考慮し、新言語の基本構造としては、従来のブロック構造を変形した木構造形式のモジュール階層構造とした。

第4章では、構造化プログラミング用言語機能の実現方式について論じ、トップダウン設計志向の段階的詳細化機能とボトムアップ設計志向のデータ抽象化機能を融合した方式を提案した。まず、両者の要となるデータ型定義機能を導入し、プログラムの段階的詳細化は、手続きとデータの両方に対して同一の手法で行えるようにした。また、データ型の定義本体の参照は、その定義を含むモジュールの下位モジュールからのみ可能とすることにより、木構造形式のモジュール階層構造の部分木の中にデータ型定義とそのデータ操作手続き群をカプセル化できるようにした。このような抽象データ型の適用範囲に柔軟性を持たせるために、定義本体の定数指定部分である配列要素数、初期値、精度などはパラメータ化できるようにした。一方、手続き定義については、プログラムの段階的詳細化によって生じる、一度だけ引用される手続きの実行効率向上をはかるため、インライン展開指示機能を設けた。また、その際、埋め込む部分をユーザ側で選択できるようにするためのコンパイル時実行機能を設け、標準パッケージに柔軟性を持たせたり、ソースレベルでの最適化ができるようにした。この

他、型チェックを徹底するための外部手続きおよびデータ型仕様明記機能や理解容易な制御構造にするための構造化コーディング機能を設けて、プログラムの高信頼化をはかった。

第5章では、このようなSPLに特徴的な言語機能を実現するための処理系の作成技法について論じ、具体的方式を提案した。第1には、段階的詳細化法によるトップダウン開発の実現、およびその各段階でのプログラムの完全な型チェックの実現のために、各モジュールの解析処理結果は共通ライブラリに保存し、他のモジュールの解析時に参照可能とすると共に、手続きのインライン展開処理は解析処理後の中間語プログラムに対して行う方式とした。第2には、このような基本処理方式の中で効率的な分割コンパイルを実現するために、各モジュールの解析処理時には、その上位環境モジュールの情報を初期環境として取り込み、あたかも自分のモジュール内の情報のように扱える方式とした。第3には、ユーザ定義データ型の同一性のチェック方法として、従来のように、データ型名による識別とデータ型定義本体の構造による識別のいずれか一方を採用すると、いずれの場合もデータ型定義機能の用途に依存した矛盾が生じることから、データ型の利用側では前者の方法、その操作手続きの定義側では後者の方法を適用するという融合方式とした。第4には、中間語レベルでの手続きのインライン展開処理において、環境情報の統合を効率的に行うために、中間語から環境情報へのポインタは、モジュール単位の先頭アドレスを管理するテーブルへのポインタと各先頭アドレスからの相対アドレスの2要素で構成しておき、インライン展開時には前者の管理テーブルだけ書き替える方式とした。第5には、SPLコンパイラは、オブジェクト言語を既存の高級言語(PCL)とするプリプロセッサ方式のため、逆ポーランド記法の中間語をインフィックス形式に変換する必要があるが、通常のスタック処理方式では、テキストの移動や編集処理が多く、効率が悪い。そこで、オペランド要素間には必ず区切り要素を1個挿入し、必要な括弧の数はこの区切り要素内に保存するような出力形式を用いて、スタック処理とテキスト操作数を大巾に削減した。

第6章では、言語処理系の性能解析を行い、共通ライブラリのアクセス方式の改良方式を提案した。まず、各フェイズのコンパイル時間を実測し、解析処理が全体の3/4を占めている一方で、展開、解釈処理は10%程度と少なく、中間語形式の手続きのインライン展開処理が比較的高速処理されていることを確認した。解析処理については、補助記憶装置との入出力時間の割合が他フェイズの2倍近くになっていることから、共通ライブラリアクセス時のページング動作解析を行った結果、ページングアルゴリズムをFIFO方式からFINUFO方式に変更し、ページサイズを512バイトから256バイトに縮小することにより、入出力時間を約50%削減できることを確めた。これらの実験結果についてはその一般性を裏づけるための理論的解析を行い、LRU方式とFIFO方式の性能差がフォールト率の小さい所で大きくなることや、LRU方式と簡易LRU方式の性能差がページ参照履歴情報の収集期間に依存していることなどを明らかにした。

第7章では、構造化プログラムのオブジェクト効率の分析を行い、効率低下要因を明らかにすると共に、各々の対策案を導いた。まず、段階的詳細化法やデータ抽象化法に基づくモジュール階層構造によって生じる、処理の冗長性の増大、モジュール間リンクエッジオーバーヘッドの発生、作業エリア

の増加などのオブジェクト効率低下要因については、ユーザ側対応策とコンパイラの最適化処理による解決策を検討した。次に、goto 文を排した構造化コーディングによって生じる、部分的処理の複写や制御変数の導入についても同様の対応策を検討した。このような分析に基づき、既存の P C L プログラムを S P L で書き直す実験を行い、所要メモリ量、実行時間共に 1 割以下の増加にとどまることを確めた。

第 8 章では、新言語 S P L の適用をより効果的にするための言語支援系の機能について論じ、具体的ツールを提案した。まず、第 1 には、オブジェクト効率の厳しい分野に構造化プログラミングを適用するために、第 1 段階では構造化に徹し、第 2 段階で性能要求に応じて会話型システムを用いて最適化を行う 2 段階プログラミング法を考案すると共に、その適用実験を行い、最適化コマンド機能を抽出した。第 2 には、S P L の有する段階的詳細化機能や構造化機能を誘導する構造エディタの基本機能について検討し、文法誤りを防止しながら効率良くプログラムの作成と修正を行う方式を具体化した。第 3 には、構造テストに用いる従来のテスト網ら基準は、品質の過大評価や冗長なテストデータ選択の欠点を有することを明らかにすると共に、その解決策として、すべての分岐に着目する従来方式に代って、パステストに本質的な分岐にのみ着目する方式およびその本質的分岐を識別する方法を導いた。そして、最後に、これらの言語支援系を有機的に結合し、1 つのプログラミング環境を構築する方式を提案した。

第 9 章では、制御用応用プログラムの信頼性と生産性向上を目的として開発された S P L が、実際の適用にあたって、その目的を達しているか否かを確めるために、適用実験および実システムへの適用効果の分析を行った。そして、従来言語の P C L に比べて、理解容易性に優れ、誤りの混入防止や早期検出に効果があること、および特に制御用応用プログラムで重要な役割を果たすタスク間共通データに関してその効果が大きいことを確めた。

以上が本論文のまとめである。本論文では、新しいプログラミング方法論をいかに具体化して実用化していくかという点に重きを置いて、そのための言語と処理系の実現方式を検討してきた。S P L コンパイラは、既に日立の制御用計算機 H I D I C 8 0 上で稼働中であり、多くの応用ソフトウェアに適用された実績を有する。本論文により得られた知見が実用的価値を有し、今後のこの分野の研究の参考となるものと信じている。

しかしながら、本研究の所期の目的であるソフトウェアの信頼性と生産性の向上は、新しい言語の適用だけで十分に達成しうるものではない。ソフトウェア開発費用の約半分がテスト工程で費されていることからわかるように、今後は、プログラムの生成、検証、修正作業を一貫して効率良く行うために、言語処理系と支援系を一体化したプログラミング環境の構築が重要と思われる。また、プログラムの誤りの多くが設計時に作り込まれるという事実から、S P L のようなプログラミング方法論を反映した言語を設計段階から使用していくような開発方式が重要と思われる。第 8 章では、これらの今後の研究課題に対する 1 つのアプローチを示したが、なお各方面からの検討が望まれる。

謝 辞

本論文をまとめるにあたり、東京大学元岡達教授に御指導と御助言を賜った。ここに衷心より感謝の意を表す。さらに、貴重な御助言をいただいた東京大学斉藤正男教授、筑波大学中田育男教授（元日立製作所）、東京大学和田英一教授に心から御礼を申し上げる。

本研究の機会を与えて下さった日立製作所コンピュータ事業本部三浦武雄本部長（元システム開発研究所）、同システム開発研究所川崎淳所長、井原廣一副所長、著者の上司として終始種々の御指導をいただいた石原孝一郎第3部部長、青山義彦第2部部長、渡辺坦主任研究員、本研究の提案方式を制御用計算機HIDIC80システムに適用して実用化する機会を与えて下さった同大みか工場桑原洋副工場長の諸氏に深く感謝の意を表す。

また、実際のシステム開発と実用化に際しては、多くの方々の御指導、御鞭撻、御助言をいただいた。日立製作所システム開発研究所野木兼六主任研究員には、SPL開発プロジェクトのリーダーとして、御指導と御討論をいただいた、同大みか工場林利弘主任技師には、本研究適用の工場側責任者として、実用的視点からの御助言、御討論をいただいた。同工場森清三技師、篠本学技師、加藤木和夫技師、同日立研究所高藤政雄研究員には、SPLコンパイラの開発を担当していただき、御討論いただいた。同システム開発研究所西垣通研究員には、ページングアルゴリズムの研究方法について御討論いただいた。同研究所山根泰章氏には、SPLコンパイラの性能評価に御協力いただいた。同研究所田中厚氏、同大みか工場高橋勇喜氏には、SPL用構造エディタの開発に御協力いただいた。これらの方々に厚く御礼を申し上げる。

最後に、本研究の遂行と本論文をまとめるにあたり、終始、温かい励ましと適切な御指導をいただいたシステム開発研究所三卷達夫主管研究長、三森定道主管研究員に深く感謝する。

参 考 文 献

- A1) AHO, A.V. and ULLMAN, J. D. Principles of compiler design. Addison-Wesley, Reading, Mass. (1977).
- A2) ARSAC, J. J. Syntactic source to source transforms and program manipulation. Comm. ACM 22, 1 (Jan 1979), 43-54.
- A3) ATKINSON, R. R., LISKOV, B. H. and SCHEIFLER, R. W. Aspects of Implementing CLU. In Proc. Annual Conf. ACM (Dec 1978), 123-129.
- B1) BALL, W. G. and ZELKOWITZ, M. V. Program complexity using hierarchical abstract computers. In Proc. NCC '78 (1978), 605-608.
- B2) BALBINE, G. Design criteria for the "structured engine". In Proc. 2nd UJCC (Aug. 1979), 422-426.
- B3) BANATRE, M., COUVERT, A., HERMAN, D. and RAYNAL, M. An experience in implementing abstract data types. Softw. -Practice and Experience 11 (1981), 315-320.
- B4) BARTH, J. M. A practical interprocedural data flow analysis algorithm. Comm. ACM 21, 9 (Sep. 1978), 724-736.
- B5) BASILI, V. R. and TURNER, A. J. Iterative enhancement: a partial technique for software development. IEEE Trans. Softw. Eng. SE-1 (Dec. 1975), 390-396.
- B6) BELADY, L. A. and KUEHNER, C. J. Dynamic space-sharing in computer systems. Comm. ACM 12 (May 1969), 282-288.
- B7) BERRY, D. M. and SCHWARTZ, R. L. Type equivalence in strongly typed languages: one more look. SIGPLAN Notices 14, 9 (Sep. 1979), 35-41.
- B8) BOEHM, B. W. Software engineering. IEEE Trans. Computer C-25, 12 (Dec. 1976), 1226-1241.
- B9) BÖHM, C. and JACOPINI, G. Flow diagrams, Turing machines and languages with only two formation rules. Comm. ACM 9, 5 (May 1966), 366-371.
- B10) BOYER, R. S., ELSPAS, B. and LEVITT, K. N. SELECT—a formal system for testing and debugging programs by symbolic execution. In Proc. Int. Conf. Reliable Software (1975), 234-245.

- C1) CAINE, S. H. and GORDON, E. K. PDL—a tool for software design. In Proc. NCC'75 (1975), 271-276.
- C2) CASHIN, P. M., JORIAT, M. L., KAMEL, R. F. and LASKER, D. M. Experience with a modular typed language: Protel. In Proc. 5th ICSE (1981), 136-143.
- C3) CHAPIN, N. Data accessibility in structured programming. In Proc. NCC'78 (1978), 597-603.
- C4) CHEN, E. T. Program complexity and programmer productivity. IEEE Trans. softw. Eng. SE-4,3 (May 1978), 187-194.
- C5) CHOW, T. S. Testing software design modeled by finite-state machines. IEEE Trans. Softw. Eng. SE-4,3 (1978), 178-187.
- C6) CHUSHO, T. and HAYASHI, T. Two-stage programming: interactive optimization after structured programming. In Proc. 3rd USA-Japan Computer Conf. (Oct. 1978), 171-175.
- C7) CHUSHO, T. A good program = a structured program + optimization commands. In Proc. IFIP'80 (Oct. 1980), 269-274.
- C8) CHUSHO, T. and HAYASHI, T. Performance analyses of paging algorithms for compilation of a highly modularized program. IEEE Trans. Sofw. Eng. SE-7,2 (March 1981), 248-254.
- C9) CHUSHO, T., et al. A language with modified block structure for data abstraction and stepwise refinement. In Proc. 3rd RIMS Symposium on the Mathematical Methods in Software Science and Engineering (Jun. 1981), 156-173.
- C10) CHUSHO, T., et al. A language-adaptive programming environment based on a program analyzer and a structure editor. In Proc. IFIP'83 (Sep. 1983), 621-626.
- C11) 中所, 野木, 高藤, 森, 加藤木, 制御用ストラクチャドプログラミング言語 SPL の処理系の特徴. 情報処理学会第17回全国大会 (1976), 5-6.
- C12) 中所, プログラムのモジュール化技法. 電子通信学会誌 62, 1 (Jan. 1979), 91-94.
- C13) 中所, 林, ページングアルゴリズムの性能に関する実験的および理論的解析. 情報処理学会論文誌 20, 6 (Nov. 1979), 460-467.
- C14) 中所, 野木, 林, 森, 段階的詳細化, データ抽象化を支援する言語 SPL のコンパイル技法. 情報処理学会論文誌 21, 3 (May 1980), 223-229

- C15) 中所, パステストに本質的な分岐に着目した網羅率尺度の提案. 情報処理学会論文誌 23, 5 (Sep. 1982), 545-552.
- C16) 中所, プログラミング言語とその会話型支援環境. 情報処理 24, 6 (Jun. 1983), 715-721.
- C17) 中所, ソフトウェアのテスト技法. 情報処理 24, 7 (Jul. 1983), 842-852.
- C18) COFFMAN, E. G. and VARIAN, L. C. Further experimental data on the behavior of the programs in a paging environment. Comm. ACM 11 (July 1968), 471-474.
- C19) COFFMAN, E. G. and DENNING, P. J. Operating system theory. Englewood Cliffs, NJ, Prentice-Hall (1973).
- C20) COHEN, E. Text oriented structure commands for structure editors. ACM SIGPLAN Notices 17, 11 (Nov. 1982), 45-49.
- D1) DAHL, D. J. and HOARE, C. A. R. Hierarchical program structures. Structured programming, Academic Press, London and New York (1972), 83-174.
- D2) DEMERS, A. J. and DONAHUE, J. E. Data types, parameters and type checking. In Proc. 7th POPL (1980), 12-23.
- D3) DENNING, P. J. Virtual memory. ACM Computing Surveys 2, 3 (Sep. 1970), 153-189.
- D4) Dept. of Defence. Requirements for high order computer language "STEELMAN". DOD, Arlington (1978).
- D5) Dept. of Defence. Reference manual for the Ada language. DOD, Washington, D. C. (1980).
- D6) Dept. of Defence. Requirements for Ada programming support environments "STONEMAN". DOD, Washington D. C. (Feb. 1980).
- D7) DeREMÉR, F. and KRON, H. Programming in-the-large versus programming in-the-small. IEEE Trans. Softw. Eng. SE-2, 4 (June 1976), 80-86.
- D8) DIJKSTRA, E. W. Goto statement considered harmful. Comm. ACM 11, 3 (March 1968), 147-148.
- D9) DIJKSTRA, E. W. The structure of the "THE"-multiprogramming systems. Comm. ACM 11, 5 (May 1968), 341-346.
- D10) DIJKSTRA, E. W. Note on structured programming. Structured programming, Academic Press, London and New York (1972), 1-82.

- D11) DIJKSTRA, E. Why naive program transformation systems are unlikely to work. Burroughs internal document EWD636 (1977).
- D12) DONALDSON, J. R. Structured programming. Datamation 19, 12 (Dec. 1973), 52-54.
- D13) DONZEAU-GOUGE, V. et al. A structure oriented program editor: a first step towards computer assisted programming. IRIA Research Report, France (Apr. 1975).
- F1) FRIEDMAN, D. P. and SHAPIRO, S. C. A case for while-until. ACM SIGPLAN Notices 9, 7 (July 1974), 7-14.
- G1) GELENBE, E. A unified approach to the evaluation of a class of replacement algorithms. IEEE Trans. Computer C-22 (June 1973), 611-618.
- G2) GESCHKE, C. M., MORRIS, J. H. Jr and SATTERTHWAIT, E. H. Early experience with Mesa. Comm. ACM 20, 8 (Aug. 1977), 540-553.
- G3) GOODENOUGH, J. B. and GERHALT, S. L. Toward a theory of test data selection. IEEE Trans. Softw. Eng. SE-1, 2 (1975), 156-173.
- G4) GRIES, D. Compiler construction for digital computers. John Wiley & Son Inc. (1971).
- G5) GRIES, D. An illustration of current ideas on the derivation of correctness proofs and correct programs. IEEE Trans. Softw. Eng. SE-2, 4 (Dec. 1976), 238-244.
- H1) HECHT, M. S. Flow analysis of computer programs. Elsevier North-Holland, New York (1977).
- H2) HENDERSON, P. An approach to compile-time type checking. In Proc. IFIP '77 (1977), 523-527.
- H3) 日立製作所, HIDIC 80 PCL言語(文法編)ソフトウェアマニュアル。(1975),
- H4) 日立製作所, HIDIC 80 SPL言語仕様編 ソフトウェアマニュアル。(May. 1977).
- H5) HOWDEN, W. E. Symbolic testing and the DISSECT symbolic evaluation system. IEEE Trans. Softw. Eng. SE-3, 4 (July 1977), 266-278.
- H6) HOWDEN, W. E. Applicability of software validation techniques to scientific programs. ACM TOPLAS 2, 3 (July 1980), 307-320.
- H7) HOWDEN, W. E. Contemporary software development environments. Comm. ACM 25, 5 (May 1982), 318-329.

- H8) HUANG, J. C. Error detection through program testing. In Current trends in programming methodology II, Prentice-Hall, New Jersey (1977).
- I1) IBM. TSO-3270 structured programming facility (SPF). Program reference manual SH20-1730-0.
- I2) ICHIBIAH, J. D. and MORSE, S. P. General concepts of the Simula 67 programming language. Annual Review in Automatic Programming 7, 1 (1972), 65-93.
- I3) ICHIBIAH, J. D., et al. Rationale for the design of the Ada programming language. SIGPLAN Notices 14, 6 (June 1979).
- I4) 池田, コンピュータユーティリティの構造, 昭晃堂, 東京 (1974).
- J1) JENSEN, K and WIRTH, N. Pascal-user manual and report. Springer (1974).
- K1) KERNIGHAN, B. W. and MASHEY, J. R. The Unix programming environment. Computer 14, 4 (Apr. 1981), 12-24.
- K2) KIEBURTZ, R. B., BARABASH, W. AND HILL, C. R. A type-checking program linkage system for Pascal. In Proc. 3rd ICSE (May 1978), 23-28.
- K3) KING, J. C. Symbolic execution and program testing. Comm. ACM 19, 7 (July 1976), 385-394.
- K4) KNUTH, D. Structured programming with goto statements, Computing surveys 6, 4 (Dec. 1974), 261-301.
- K5) 久保, 他5名, 列車運行管理システムへのストラクチャードプログラミングの適用. 第14回「鉄道におけるサイバネティクス利用」国内シンポジウム (Nov. 1977).
- L1) LAMPSON, B. W. et al. Report on the programming language Euclid. SIGPLAN Notices 12, 2 (Feb. 1977).
- L2) LAUER, H. C. and SATTERTHWAITTE, E. H. The impact of Mesa on system design. In Proc. 4th ICSE (1979), 174-182.
- L3) LEBLANC, R. J. Extensions to Pascal for separate compilation. ACM SIGPLAN Notices 13, 9 (1978), 30-33.
- L4) LEBLANC, R. J. and FISCHER, C. N. On implementing separate compilation in block-structured languages. In Proc. ACM Symp. Compiler Construction (Aug. 1979), 139-143.
- L5) LEBLANG, D. B. Abstract syntax based programming environments. In Proc. AdaTEC Conf. on Ada (Oct. 1982), 187-200.

- L6) LISKOV, B. and ZILLS, S. Programming with abstract data types. ACM SIGPLAN Notices 9, 4 (Apr. 1974), 50-59.
- L7) LISKOV, B., SNYDER, A., ATKINSON, R. AND SCHAFFERT, C. Abstraction mechanism in CLU. Comm. ACM 20, 8 (Aug. 1977), 564-576.
- L8) LISKOV, B. and ZILLS, S. An introduction to formal specifications of data abstractions. In Current trends in programming methodology I, Prentice-Hall, New Jersey (1977), 1-32.
- L9) LISKOV, B. et al. CLU Reference Manual. MIT/LCS/TR-225, M. I. T., Cambridge, Mass. (Oct. 1979).
- L10) LOMET, D. B. Data flow analysis in the presence of procedure calls. IBM J. Res. Develop. 21, 6 (Nov. 1977), 559-571.
- L11) LOVEMAN, D. B. Program improvement by source-to-source transformation Journal of ACM 24, 1 (Jan. 1977), 121-145.
- L12) LUCAS, H. C. Jr and KAPLAN, R. B. A structured programming experiment. The Computer Journal 19, 2 (May 1976), 136-138.
- M1) 益田, 他2名, ページングマシンにおけるスワッピングアルゴリズムの比較とプログラムの動作解析。 情報処理 13, 2 (1972), 81-88.
- M2) 益田, 亀田, オペレーティングシステムの性能解析, 情報処理学会, 情報処理業書9 (1982)。
- M3) MATTSON, R. L. et al. Evaluation techniques for storage hierarchies. IBM Syst. J. 9 (1970), 78-117.
- M4) McCABE, T. J. A complexity measure. IEEE Trans. Softw. Eng. SE-2, 4 (Dec. 1976), 308-320.
- M5) McCLORE, C. L. Top-down, bottom-up and structured programming. In Proc. 1st ICSE (Sep. 1975), 89-94.
- M6) McCLORE, C. L. A model for program complexity analysis. In Proc. 3rd ICSE (May 1978), 149-157.
- M7) McCOWAN, C. Structured programming: review of some practical concepts. Computer 8, 6 (June 1975), 25-30.
- M8) MEDINA-MORA, et al. An incremental programming environment. IEEE Trans. Softw. Eng. SE-7, 5 (Sep. 1981), 472-482.
- M9) MILLER, E. F. Jr and LINDAMOOD, G. E. Structured programming: top-down approach. Datamation 19, 12 (Dec. 1973), 55-57.

- M10) MILLER, E. F. Program testing : art meets theory. Computer 10, 7 (1977), 42-51.
- M11) MILLER, E. F. ,et al. Tutorial : software testing and validation techniques. IEEE Catalog EH0138-8 (1978).
- M12) MITCHELL, J. G. ,MAYBURY, W. and SWEET, R. Mesa language manual. CSL-78-1, Xerox PARC, Palo Alto, Calif. (Feb. 1978).
- M13) MORRIS, J. M. ,et al. The design of a language-directed editor for block-structured language. ACM SIGPLAN Notices 16, 6 (Jun. 1981), 28-33.
- M14) MYERS, G. J. Reliable software through composite design. Petrocelli/Charter, New York (1975).
- M15) MYERS, G. J. Software reliability. John Wiley and Sons Inc. (1979).
- M16) MYERS, G. J. The art of software testing. A Wiley-Interscience, New York (1979).
- N1) NAKAJIMA, R. ,et al. The Iota programming system. In Proc. IFIP '80 (1980), 299-304.
- N2) 中田, コンパイラの技法. 竹内書店 (1971)。
- N3) 日本規格協会, 電子計算機プログラム用言語 ALGOL (水準 7000)。
JIS C 6210-1972 (Mar. 1972)。
- N4) 野木, 他 3 名, 階層化プログラミング言語 SPL の評価, 情報処理学会ソフトウェア工学研究会資料 9-1 (1979)。
- O1) 緒方, 他 2 名, OS 7 メモリスケジューリング方式。 情報処理学会 第 14 回全国大会 (1973), 363-364
- P1) PARNAS, D. L. On the criteria to be used in decomposing systems into modules. Comm. ACM 15, 12 (Dec. 1972), 1053-1058.
- P2) PARNAS, D. L. Designing software for ease of extension and contraction, In Proc. 3rd ICSE (May 1978), 264-276.
- P3) PARNAS, D. L. and SIEWIOREK, D. P. Use of the concept of transparency in the design of hierarchically structured systems. Comm. ACM 18, 7 (July 1975), 401-408.
- R1) RAPPS, S. and WEYUKER, E. J. Data flow analysis techniques for test data selection. In Proc. 6th ICSE (1982), 272-278.
- R2) RITCHIE, D. M. and THOMPSON, K. The Unix time-sharing system. Comm. ACM 17, 7 (July 1974), 365-375.

- R3) RODRIGUES-ROSELL, Empirical data reference behavior in data base systems. Computer 9 (Nov. 1976), 9-13.
- S1) SANDEWALL, E. Programming in an interactive environment: the "Lisp" experience. ACM Computing surveys 10, 1 (Mar. 1978), 35-72.
- S2) SANFIELD, S. H. The scope of variable concept: the key to structured programming. ACM SIGPLAN Notices 9, 7 (July 1974), 22-29.
- S3) SCHEIFLER, R. W. An analysis of inline substitution for a structured programming language. Comm. ACM 20, 9 (Sep. 1977), 647-654.
- S4) SCHUSTER, E. A., OZKARAHAN, E. A. and SMITH, K. C. A virtual memory system for a relational associative processor. In Proc. NCC '76 (1976), 855-862.
- S5) SELF, L. E. and DAVIS, E. W. Analyzing the modular design of Fortran programs. In proc. COMPSAC '77 (1977), 554-560.
- S6) SHAW, M. et al. validating the utility of abstraction techniques. In Proc. Annual Conf. ACM (Dec. 1978), 106-110.
- S7) SHERMAN, S. W. and BRICE, R. S. performance of a database manager in a virtual memory system. ACM Trans. Database Syst. 1 (Dec. 1976), 317-343.
- S8) SNOWDON, R. A. PEARL: an interactive system for the preparation and validation of structured programs. Tech. Rep. Series 28, Univ. Newcastle, Computing Lab. (Nov. 1971).
- S9) STALLMAN, R. M. EMACS, the extensible, customizable, self-documenting display editor. ACM SIGPLAN Notices 16, 6 (Jun. 1981), 147-156.
- S10) STANDISH, T. A., KIBLER, D. F. and NEIBORS, J. M. Improving and defining programs by program manipulation. In Proc. ACM annual conf. (1976), 509-516.
- S11) STANDISH, T. A., et al. The Irvine program transformation catalogue. Dept. Inf. and Computer Science, UCI (1976).
- S12) STANDISH, T. A. The importance of Ada programming support environments. In Proc. NCC '82 (1982), 333-339.
- S13) STEVENS, W. P., MYERS, G. J. and CONSTANTINE, L. L. Structured design. IBM Syst. J. 13, 2 (1974), 115-139.
- S14) SYSTEM5, プログラムと "go to". 数学セミナ 11, 8 (Aug. 1972), 38-40.

- T1) TEITELBAUM, T. , et al. The Cornell program synthesizer : a syntax - directed programming environment. Comm. ACM 24, 9 (Sep. 1981), 563-573.
- T2) TEITELMAN, W. , et al. The Interlisp programming environment. Computer 14, 4 (Apr. 1981), 25-32.
- T3) TOMPSON, E. W. and BRIDGE, R. F. A module interface specification language. In Proc. 12th Design Automation Conf. (June 1975), 42-49.
- T4) TOWSTER, E. A convention for explicit declaration of environments and top-down refinement of data. IEEE Trans. Softw. Eng. SE-5, 4 (July 1979), 374-386.
- V1) VOGES, U. , et al. SADAT-an automated testing tool, IEEE Trans. Softw. Eng. SE-6, 3 (1980), 286-290.
- W1) 和田, エディタとテキスト処理, 構造エディタ(2) Pascalの場合. bit 15, 2 (Feb. 1983), 185-192.
- W2) WASSERMAN, A. I. A top-down view of software engineering. In Proc. 1st ICSE (1975), 1-7.
- W3) WATERS, R. C. Program editors should not abandon text oriented commands. ACM SIGPLAN Notices 17, 7 (July 1982), 39-46.
- W4) WELSH, J. , SNEERINGER, W. J. AND HOARE, C. A. R. Ambiguities and insecurities in Pascal. Software-Practice and Experience, 7, 6 (Nov. -Dec. 1977), 685-696.
- W5) WEYUKER, E. J. and OSTRAND, T. J. Theories of program testing and the application of revealing subdomains. IEEE Trans. Softw. Eng. SE-6, 3 (1980), 236-246.
- W6) WHITE, L. J. and COHEN, E. I. A domain strategy for computer program testing. IEEE Trans. Softw. Eng. SE-6, 3 (1980), 247-257.
- W7) WIRTH, N. Program development by stepwise refinement. Comm. ACM 14, 4 (April 1971), 221-227.
- W8) WIRTH, N. The programming language PASCAL. Acta Informatica 1 (1971), 35-63.
- W9) WIRTH, N. Systematic programming : an introduction. Prentice-Hall (1973).

- W10) WIRTH, N. Modula : a language for modular multiprogramming. Softw.-Practice and Experience 7, 1 (1977), 3-35.
- W11) WULF, W. A. , et al. BLISS: a language for systems programming. Comm. ACM 14, 12 (Dec. 1971), 780-790.
- W12) WULF, W. A. Alphard : toward a language to support structured programs. NTIS (April 1974)
- W13) WULF, W. A. , LONDON, R. L. and SHAW, M. An introduction to the construction and verification of Alphard programs. IEEE Trans. Softw. Eng. SE-2, 4 (Dec. 1976), 253-265.
- W14) WULF, W. A. and LONDON, R. A. Abstraction and verification in Alphard: defining and specifying iteration and generators. Comm. ACM 20, 8 (Aug. 1977), 553-564.
- Y1) YOSHIZAWA, Y. and KONDO, M. Effect of partially preloaded program on paged virtual memory. Journal of information processing 1, 2 (1978), 92-97.
- Z1) ZELKOWITZ, M. V. and BALL, W. G. Optimization of structured programs. Softw.-Practice and Experience 4 (1974), 51-57.
- Z2) ZELKOWITZ, M. V. Perspectives on software engineering. ACM Computing Surveys 10, 2 (June 1978), 197-216.