

第 9 章 適用効果

第 9 章 適用 効果

9.1 概 要

S P L 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プログラミング言語である。そのため、これまで述べてきたように、従来言語の P C L には無い新機能が豊富に導入されている。

そこで、本章では、これらの新機能を有する S P L が、実際の適用にあたって、所期の目的を達成しているか否かを確かめるために、次の 2 つの方法でその評価を行った。まず、第 1 には、P C L 経験者 20 名を対象に S P L の適用実験を行い、P C L と比較しながら S P L の記述性と保守性の評価を行った。第 2 には、S P L の最初の適用プロジェクトで検出された誤りおよび工数について、P C L プログラムの場合との比較分析を行った。

9.2 実験と評価

9.2.1 実験の目的

プログラムの生産性への影響が大きいと思われるプログラミング言語の記述性と保守性について、新言語 S P L と従来言語 P C L を実験的に比較評価する。

9.2.2 実験の方法

まず、被験者として、P C L プログラムの開発に 2 ~ 3 年の経験を有するプログラマ 20 名を選んだ。そして、ある問題を P C L で解くというプログラミングテストを行い、その成績に基づいて、プログラム作成能力が同程度になるように 10 人ずつの 2 つのグループに分けた。各々を S P L グループと P C L グループと呼ぶ。これらの被験者は S P L については全く未経験であるため、1 日 2 時間の S P L 教育を 5 日間行った。

(1) 記述性の評価

以下の 3 種類の問題について、機能仕様書、G F C (General Flow Chart)、テーブル仕様書を与え、一定時間内に各々のグループの言語でプログラムを作成させ、各グループの平均点を測定する。

問題 1 : N 人の得点が格納されているテーブルをもとに、平均点が同一になるようにグループ分けする。

問題 2 : 氏名と電話番号をカードリーダーから入力し、アルファベット順にラインプリンタに出力する。

問題 3 : 部品倉庫から、部品の入出庫を行う。

(2) 誤り検出容易性の評価

以下の問題について、機能仕様書、G F C、テーブル仕様書、および各々のグループの言語で

記述したプログラムリストを与え、一定時間内に複数個の誤りを検出、修正させ、各グループの平均点を測定する。

問題 4：部品組立て作業のベルトコンベアを制御する。

(3) 理解容易性の評価

以下の問題について、不完全な G F C およびテーブル仕様書と各グループの言語で記述したプログラムリストを与え、一定時間内に詳細機能を判読させ、各グループの平均点を測定する。

問題 5：機械加工作業の進捗管理を行う。

(4) 変更容易性の評価

上記の問題 5 のプログラムを説明し、内容を理解させた後、テーブルのデータ構造およびその操作方法の変更を指示し、一定時間内にプログラムを修正させ、各グループの平均点を測定する。

9.2.3 実験結果と考察

実験結果を表 9.1 に示す。まず、記述性については、3 回実験したが、いずれも S P L の方が P C L よりも劣った。これは 10 時間程度の S P L の教育では、未だ十分な理解がなされていないためと思われる。しかしながら、実験の回数を重ねるに従って、P C L との差が縮まっていることから、S P L の習熟時には、P C L と同程度以上になることが期待される。

一方、書き易さよりも読み易さを重視する構造化プログラミングの観点からは、理解容易性およびそれに基づく誤り検出容易性や変更容易性などの保守性が重要な評価基準となるが、この点についてはいずれの実験でも S P L の方が優れているという結果が得られた。

9.3 実システムへの適用効果

次に、S P L を最初に適用したプロジェクトを例にとり、実システムへの適用効果を調べた。このプロジェクトで開発した制御用応用プログラムの規模は全体で 130 K 語（1 語は 16 ビット）であり、そのうちの 40 K 語を S P L、残りの 90 K 語を P C L で記述している。

その結果、まず、S P L 適用の定性的効果として、次のような項目が掲げられた。

- (1) 環境モジュールで宣言する共通データの設計が完了しないと処理モジュールのプログラムをコンパイルできないことから、必然的に事前に十分な設計が行われ、設計不良による後戻り作業の発生が少なくなる。
- (2) ソースプログラムリストのドキュメント性が向上し、机上検証が容易になるため、誤りを早期に検出できる。
- (3) コンパイラによるモジュール間インターフェイスやデータ型のチェックが厳しいため、誤りを早期に検出できる。
- (4) 共通データは階層構造の環境モジュール群により一元管理されること、および手続きやユーザ定義データ型のインターフェイスの検証機能により、仕様変更に伴うプログラム変更を漏れなく

表9.1 SPLとPCLの比較評価実験の結果

評価基準		解答 時間	SPLグループ 平均点	PCLグループ 平均点	SPL/PCL
大項目	小項目				
記述性	問題1	2時間	58.3	76.4	0.76
	問題2	2	71.0	80.0	0.89
	問題3	4	77.5	85.0	0.91
保守性	誤り検出容易性	2	85.0	80.0	1.06
	理解容易性	2	60.0	54.6	1.10
	変更容易性	2	79.6	64.0	1.24

行うことができる。

一方、SPL適用の定量的効果として、誤りの混入防止と早期検出に関する評価データを収集、分析した結果、次のような項目が明らかになった。

- (1) SPLプログラムの誤り発生率はPCLの場合の59%に減少しており、誤りの混入防止に効果がある。
- (2) センターテストおよび実機テストで検出した誤りの数を比較すると、図9.1に示すように、センターテストでの誤り検出率が、PCLの場合には56%であったが、SPLでは80%にまで向上しており、誤りの早期検出に効果がある。

なお、ここで、センターテストとは、計算機センタに設置してあるターゲットマシンを用いた単体テスト、結合テストを意味する。また、実機テストは顧客納入用マシンを用いたシステムテストを意味し、各種のプロセス入出力機器またはそのハードウェアシミュレータを接続し、システムの稼動状態とほぼ同じ状態でテストを行う。

9.4 適用実績

SPLの開発は、1975年春に開始され、コンパイラの第1版は1977年3月に完成した。その後、78/7、79/6、80/5、82/7に4度の改訂が行われ、今日に至っている。このうち、第4版までは日立の制御用計算機HIDIC80(16ビットマシン)で稼動するものであり、最新の第5版は、その後継機種であるHIDIC V90(32ビットマシン)で稼動する。

その間にSPLを適用したシステムの数、1983年11月現在の時点で約370件のほり、総ステップ数は約9040キロステップに達している。なお、これは、HIDIC80システムを主力製品とする日立製作所大みか工場の実績であるが、この他、社内他事業所に4件、顧客に210件のSPL納入実績を有している。

9.5 結 言

SPLの適用実験および実システムへの適用を通じて、その効果を調べた。その結果、SPLは、従来方式に比べて、理解容易性に優れており、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。このことから、制御用応用プログラムの信頼性と生産性の向上という、SPLの開発目的はほぼ達成されたと思われる。

しかしながら、ここで行った定量的評価は未だ十分なものとは言えず、今後も種々の面からの評価データを収集、分析していく必要がある。なお、評価実験では、記述性に関しては良い結果が得られなかったが、言語の高機能化により言語仕様が複雑化し、習得が難しくなるという問題は極力避けなければならない。また、SPLの言語仕様およびその基礎となるプログラミング方法論についての効果的な教育方法は今後の課題である。

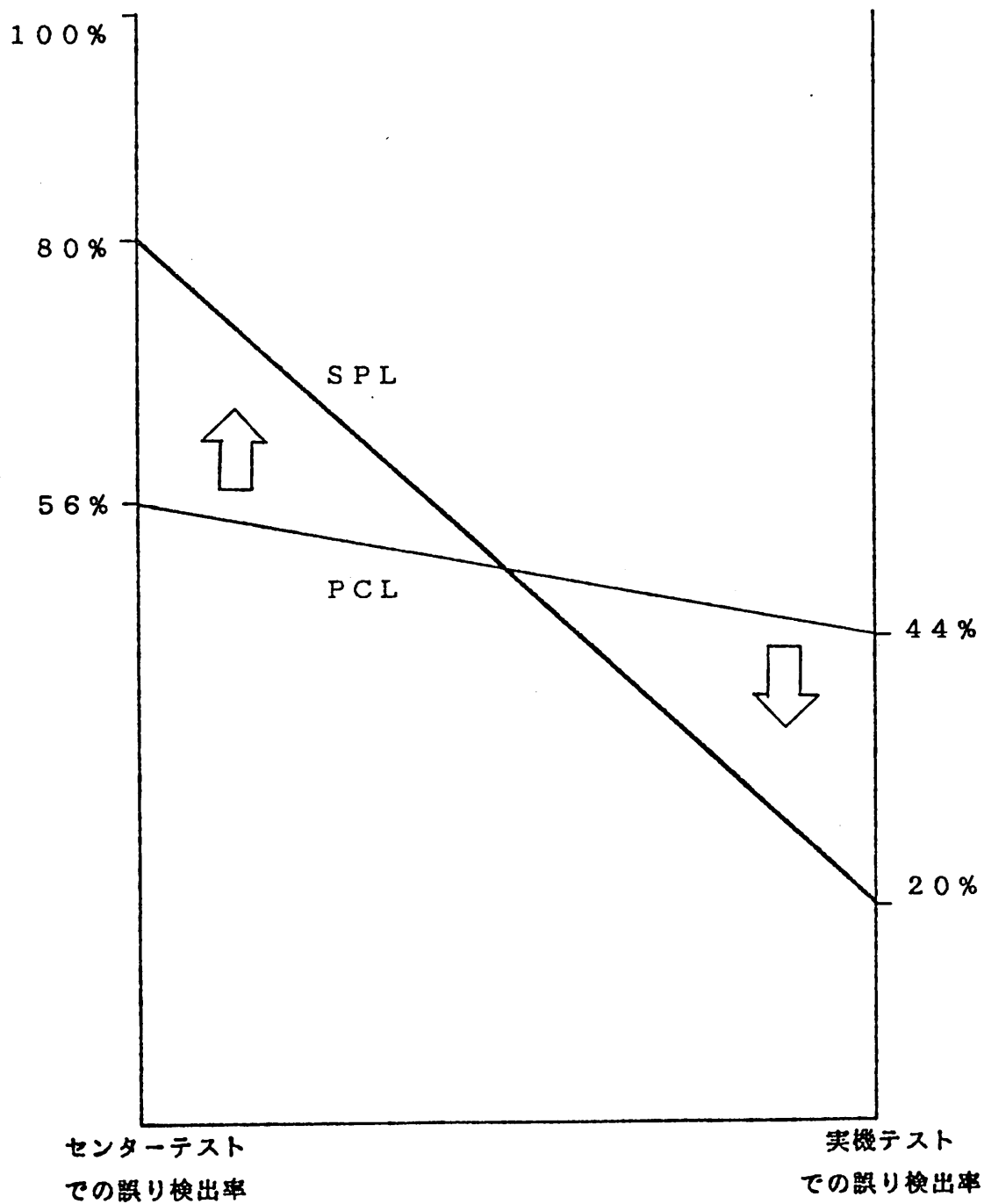


図9.1 センターテストと実機テストの誤り検出比率