

自動車用電子機器プログラムの自動生成システム－A P E S

Automatic Placement and Evaluation System of Assemble Program used for Car Electronics Devices

葛 崎 偉⁽¹⁾
Qi-Wei Ge

池 添 朗⁽²⁾
Akira Ikezoe

斗 納 宏 敏⁽³⁾
Hirotohi Tonou

要 旨

地球環境問題への対応や安全に対する社会的ニーズの高まり等を背景に、自動車用電子制御機器は高機能化・多様化し、制御用マイコンのプログラムは複雑化・大規模化してきた。これに伴い、プログラムの開発は困難となり工数も増大している。プログラム開発を効率よく行うために、プログラム自動生成システム－A P E Sを開発した。

本システムは、仕様書に従って標準化したプログラム部品（モジュール）を作成プログラムに配置し、仕様書に記述されている制御データを作成プログラムに書き込むことにより、アセンブル・プログラムを自動的に生成するものである。システムのねらいは、(1)プログラムの信頼性を向上させ、(2)プログラムの設計工数を削減することにある。

Recently, in order to comply with the increasing demand for the environment conservation issue and the driving safety etc., car electronics devices have been required to possess more up-graded and more diversified functions, and programs of the microprocessors used in the devices have become unprecedentedly large-scaled and complicated. So that, developing such programs has become more difficult and cost much time. To support designing the programs, we have developed a programming support system APES (Automatic Placement and Evaluation System of assemble program).

In this system, an assemble program is automatically produced, according to given specification, by composing prepared program parts (modules) and transcribing required constants from the specification into the program. This system aims at (i) enhancement of the program reliability and (ii) decreasing of the program developing cost.

1. ま え が き

近年、地球環境問題への対応や安全に対する社会的ニーズの高まり等を背景に、自動車用電子制御機器は高機能化・多様化し、制御用マイコンのプログラムは複雑化・大規模化してきた。これに伴い、プログラムの開発は困難となり工数も増大しており、プログラムの生産性の向上や信頼性の確保が急務となっている。現行の開発手段ではその対応に限界があり、新たな開発環境が必要である。

ソフトウェアの生産性と信頼性の向上策として、CASE (Computer Aided Software Engineering) が注目されている。¹⁾ CASEは、要求仕様の分析からプログラム生成までの開発を支援するツールのことであり、事務処理システムを対象とした分野で特に発達しており、²⁾ 制御システムを対象とした分野でもTeamwork、CARDtools、AutoCodeなどが商品化されている。³⁾ 最近では、ソースコードの自動生成を支援するものも現れてきた。

しかしながら、市販のCASEを自動車用電子制御機器のプログラム開発に適用するには、いくつかの問題点がある。

自動車用電子制御機器のプログラムは、ハードウェアに密着しており、高速のリアルタイム制御が要求されているので、処理スピードの面で汎用CASEでの対応は困難である。また、コストを抑えるために限られたメモリでのプログラム設計が必要であり、アセンブル言語を採用しているものが多いため、高級言語を対象としたCASEは利用できる範囲が狭い。

これらの理由から、当社独自にCASE開発を行っており、その一環として自動車用電子制御機器のプログラムを対象としたアセンブル・プログラム自動生成システム—APES (Automatic Placement and Evaluation System of assemble program) の開発を行った。本論文ではAPESのシステム構成と構築技術を中心に紹介する。

2. 開発構想

従来からソフトウェアの開発^{4) 5)}は、要求分析、

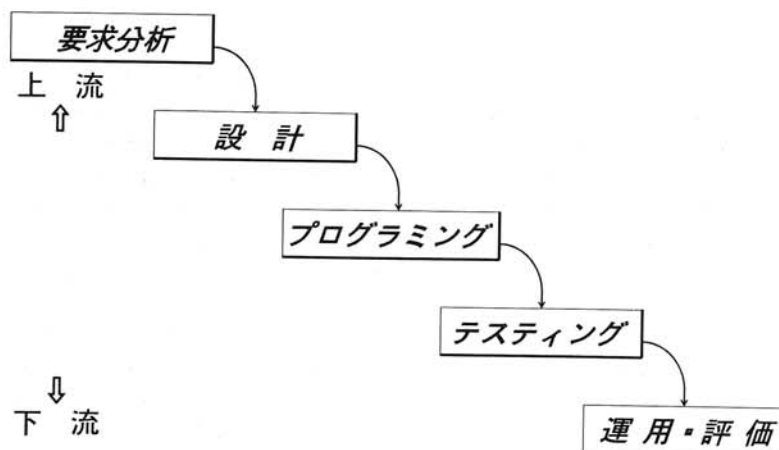


図-1 ウォータフォールモデル
Fig. 1 Waterfall Model

設計、プログラミング、テスト、運用・評価の各工程を順に逐次進める形をとっている。このような開発のモデルを、滝の流れに例えて「ウォーターフォールモデル」(Waterfall Model) (図-1)と呼ぶ。このようなソフトウェアの開発過程においては、ユーザの要求仕様を頂点とした「トップダウン・アプローチ」を取り、上流から下流へと厳密に各工程を完遂しながら進めていくことを前提としている。

開発したプログラムが要求仕様通りの処理をしているかを確認する作業をデバッグと呼び、デバッグの結果、要求仕様に反した動作をしていれば上流の工程に戻ってプログラムを修正することになる。下流での不具合の発見は、上流に逆上って修正する必要を生じるため、下流に行けば行くほど修正作業は困難となる。最終的にシステムとしての品質を保証するため、この作業は要求仕様通りの動作をするまで繰り返され、プログラムの開発工数において大きなウェイトを占めている。特に、自動車制御では小さなミスが重大な事故につながる可能性があるため、プログラムの動作確認は特に厳密に行う必要があり、全工数の50%以上をデバッグに費やしている。⁶⁾

自動車用電子制御機器のプログラムのデバッグは、主に、

- ① プログラムの机上確認
- ② シミュレータ・ベンチによる動作チェック
- ③ 自動車への装着状態での動作チェック

で行われているが、このデバッグ方法は作業効率が高く、バグを突き止めるのに多大な時間を要している。

当社では、「CAPAS」(プログラム自動解析システム)⁶⁾・「ADS」(自動デバッグシステム)⁷⁾等プログラム・デバッグの支援システムの

開発を行っており、作業の効率化、信頼性の向上を図ってきた。しかし、制御定数のデータ入力ミスや仕様・データ変更の見落としなどのバグの修正のため、各工程間でのフィードバックを何度となく繰り返して設計を行っているのが現状である。制御内容の多様化、システムの大規模化・複雑化に伴い、プログラムの規模が大きくなりつつある今日、設計工程の効率化を図ることが必要である。

このため、仕様書からプログラムの自動生成を行うことで、工程の上流での設計品質の確保と、プログラムの開発期間の短縮を図ることを目的として、「自動車用電子制御機器プログラムの自動生成システム－APES」の開発を行った。

3. APESの概要

当社ではプログラムを機能単位毎に細かく分割したものをモジュールと呼んでいる。また、使用実績があり仕様が安定したものを標準モジュールと呼び、再利用することでプログラム設計の効率化を図っている。APESは、この標準モジュールを使った設計を更に合理化したもので、仕様書に従ってプログラムで使用するモジュールを自動的に配置するシステムである。



図-2 システムの外観

Fig.2 Outer view of APES

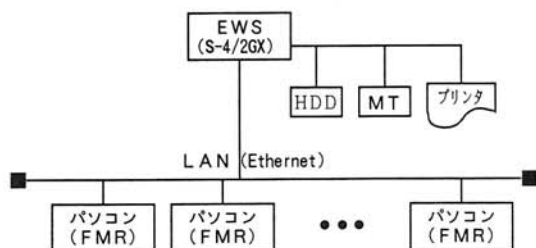


図-3 ハードウェア構成図

Fig.3 Hardware structure of APES

本章では、APESのシステム構成とシステム機能の実現手法を紹介する。図-2にシステムの外観を示す。

3.1 システム構成

APESのハードウェアは、EWS (Engineering Work Station) とその周辺機器 (ハードディスク、MT装置、プリンタ) とパソコン端末から構成され、EWSとパソコン端末をLANでつないでいる (図-3)。

APESの機能構成は、

- ① 仕様書データ抽出 (本文での説明を省略)
- ② プログラム自動生成
- ③ 仕様書データ照合

からなっている (図-4)。

APESは、EWSをホストとし、パソコン端末から複数の設計者が同時に使えるように設計し

ている。

3.2 プログラムの自動生成

図-5は、プログラムの自動生成のソフトウェア構成を示したものである。

制御用アセンブル・プログラムは、一般に、

- ① RAM定義部
- ② 制御定数定義部
- ③ 制御ロジック部
- ④ 共通サブルーチン部

の四つのブロックから構成されている。APESはそれぞれのブロックを標準モジュールライブラリと電算化した仕様書より抽出したデータから、自動的に生成する。

標準モジュールライブラリは、制御ロジックを記述したアセンブルコード部と自動生成に必要な付随情報部から構成されている。付随情報には、RAMに割り付けられる変数の定義と制御定数の記述フォーマットの定義が書かれている。APESでは、仕様書に記述された要求機能に応じて、必要な標準モジュールを再結合してRAM定義部、制御定数定義部、制御ロジック部の各ブロックを生成する。共通サブルーチン部は、機種毎に必要なサブルーチンを共通部品ライブラリから取り出して配置する。

本節では、プログラムの各ブロックの生成手法

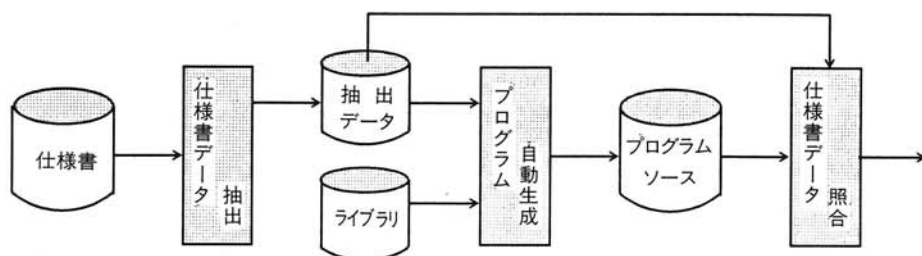


図-4 APESの処理の流れ

Fig.4 Processing flow of APES

について紹介する。

3. 2. 1 制御ロジック部の生成

当社において用いられているモジュールは、経済性を考慮して、プログラムサイズが小さくなるように設計しており、機能単位毎に分割してはいるが、モジュール相互の依存度合が強く、独立して用いることが少ない。その特性により、各々のモジュールの配置位置によって正常に動作しない場合がある。このため、モジュールを配置する際に、配置の相対位置を考慮して行わなければならない。本システムでは、

① モジュールを先頭または最後に置く
 ② 他のモジュールよりも前または後に置く
 ③ 他のモジュールの直前または直後に置く

といった「配置順序」情報や「関連配置」情報（配置したモジュールから呼び出すサブルーチンなど、そのモジュールを使用する際に必ず配置されるべきモジュール）をモジュール配置の「関連情報」として予めライブラリに登録し、モジュール

ル配置を行う時にその情報を参照する。図-6にその記述例およびそれぞれの記号説明を示す。

例えば、仕様書から次のモジュールの配置が要求されているとする。

- ・ モジュール BEGIN
- ・ モジュール A
- ・ モジュール B
- ・ モジュール C
- ・ モジュール D
- ・ モジュール E
- ・ モジュール END

図-6の「関連配置」情報（モジュール C [モジュール X]）より、上記のモジュール以外に「モジュール X」も配置しなければならないことが分かる。そこで、標準モジュールライブラリから上記のモジュールと「モジュール X」を取り出し、図-6の「配置順序」に従ってモジュールの配置をすれば、制御ロジック部が生成される（図-7）。その他、モジュールによっては多くのプログラム生成

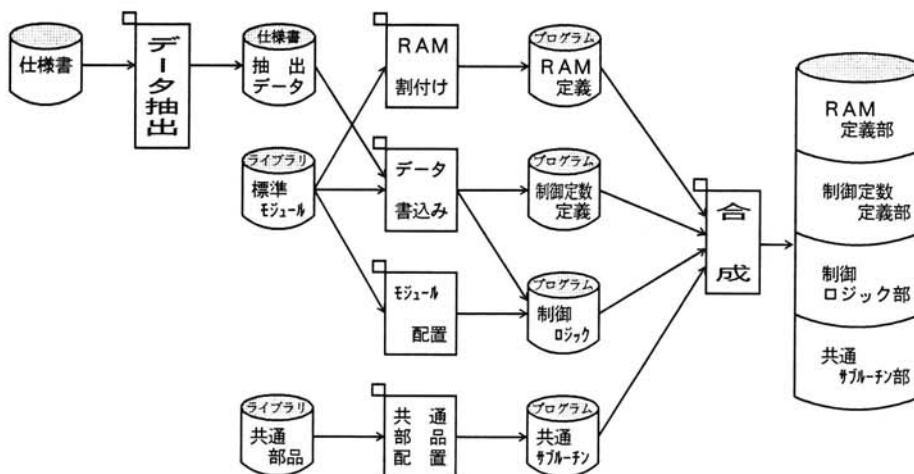


図-5 「プログラム自動生成」のソフトウェア構成図

Fig.5 Process diagram of automatic program generation

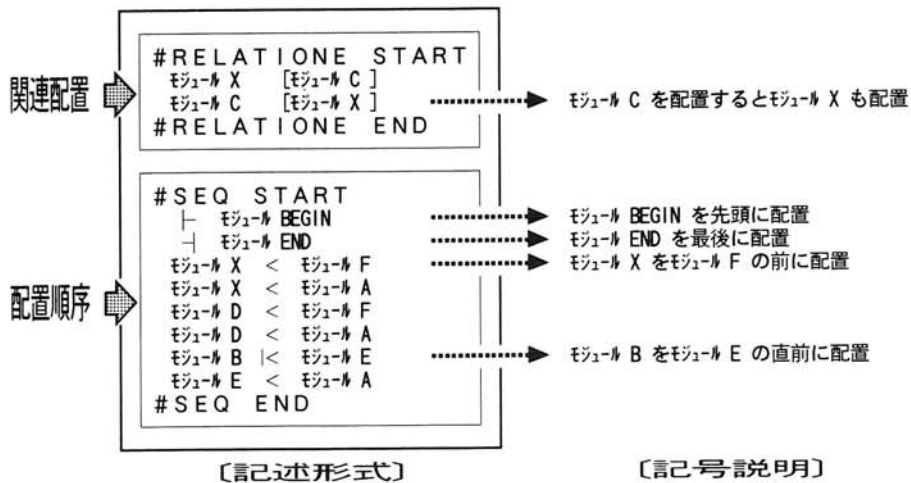


図-6 モジュール配置の関連情報の例

Fig.6 An example of related information for module placement

に展開するため、機種毎に変更可能な制御パラメータを設定しており、この場合は仕様書から制御パラメータの書き込みも同時に行う。

「配置順序」に特に規定されていないモジュール、例えば例の「モジュール C」はどこに配置してもよい。上記の例の場合は「モジュール C」を「モジュール X」の前に配置した(図-7)。

3. 2. 2 制御定数定義部の生成

制御定数とは、エンジンのパラメータに応じて決まる制御量の2次元や3次元のデータであり、例えば、エンジン冷却水温に応じて制御量を変化させる2次元のテーブルデータ(図-8)がある。制御定数定義部はプログラム中の各モジュールにおいて使用される制御定数の集まりである。プログラムにおける各制御定数のフォーマットはそれ

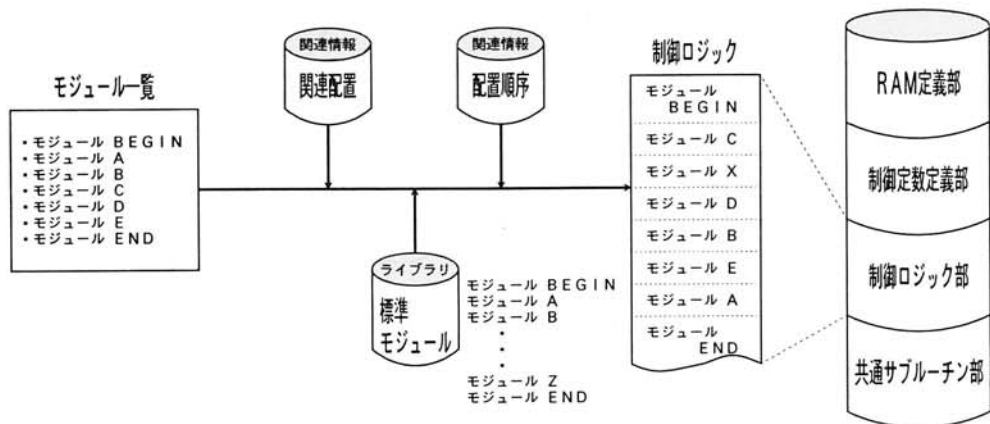


図-7 制御ロジック部の生成例

Fig.7 An example of logic block of a program

それぞれの標準モジュールの付随情報に書いてある制御定数記述フォーマットに従う。

APESでは、使用する標準モジュールの付随情報から制御定数記述フォーマットを読み出して、仕様書に記述された制御定数を所定の位置に書き込むことによって制御定数定義部を生成する。その他、仕様変更によるモジュールおよび関連部分の追加や削除を行い易くするため、各モジュールの複数の制御定数を識別子を付けて、一箇所にまとめて配置している。

3. 2. 3 RAM定義部の生成

プログラムにおける変数は、プログラム実行中の演算結果や制御量などを表すものであり、データはRAMに格納している。プログラムのRAM定義部は、制御ロジック部に配置される各モジュールの変数をRAMに割り付ける部分である。

APESでは、標準モジュールの付随情報から変数定義を読み出して、変数のRAM割り付けを行っている。制御用マイコンでのRAMは、ビット操作やメモリ保持などの機能によって複数の領域に別れており、それぞれの変数はこれらの特定のRAMアドレス領域内に割り付けを行わなければならない。また変数によっては複数のモジュール

に使用されることもあり、同じ変数の重複割り付けを避ける必要がある。APESでは、これらのRAM割り付け上の制約を守ってRAM定義を行っている。

3. 2. 4 共通サブルーチン部の生成

共通サブルーチンはプログラム間で共用可能なプログラム部品であり、プログラム実行時にモジュールから呼び出されるものである。

共通サブルーチンの使用については、仕様書に明記されない場合が多い。このため、APESでは共通サブルーチンとそれを使用したプログラムの機種情報をライブラリに共通部品として登録しており、仕様書に記述された機種情報を利用して共通部品から該当する共通サブルーチンを取り出して配置するようにしている。

3. 3 仕様書データとの照合

APESはデバッグの機能として、仕様書に記述してある制御定数とプログラム中の制御定数を比較する機能を持っている。本来ならば、APESで生成されたプログラム中の制御定数は仕様書に記述してある制御定数と一致するはずであるが、標準モジュールの使用率が100%に達していないことや、設計変更によるプログラムの部分修正が

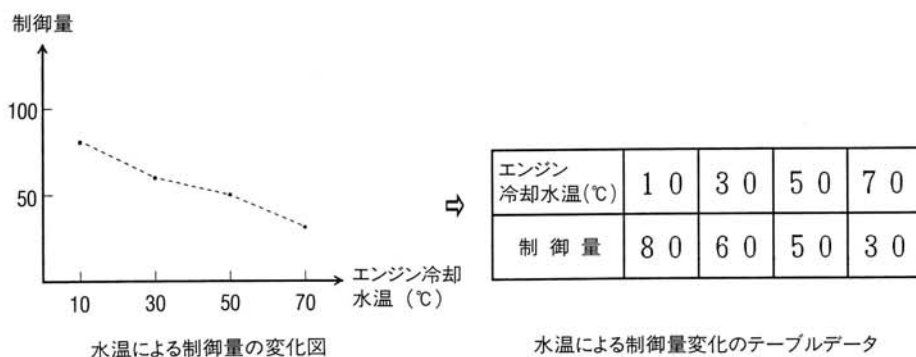


図-8 制御定数の説明例

Fig. 8 An example of control constant

発生することなどから、最終的にデータ不一致が生じる可能性がある。そのために、誤ってデータを書き込んだり書き換えたりしても検出できるよう本機能をシステムに持たせた。

データ照合の結果は以下の二種類のフォーマット

- ① 一致しないデータのみ
- ② 両方のデータ一覧

で出力する。前者・後者共にデバッグ情報として利用し、後者はドキュメント資料としても利用できる。

3. 4 プログラムの修正機能

A P E S は部品化したプログラムの再利用を前提とし、標準モジュールの合成によりプログラムを生成するシステムである。従って、新しくプログラムを設計する場合、プログラム部品の再利用率が高ければ高いほど生産性も向上することになる。ところが、自動車用電子制御機器のプログラム設計では、最初から設計するより以前設計したプログラムの制御定数を変更したり、他機種のプログラムを一部流用したりすることが多く、モジュールの合成だけで生産性向上は望めない。実際の設

計工程での効率化を図るには、一度作成したプログラムの設計変更が簡単に行えることが必要である。A P E S ではこうした場合にも対応可能とするため、プログラムの設計変更を行う機能を持たせている。

A P E S では一度作成したプログラムに設計変更を行う場合、変更を指示する仕様書とプログラムのモジュール構成を比較し、自動的にモジュールの追加・削除・定数変更を行う機能を有している。このため、プログラムの部分的な変更作業においても効率化が可能である。

4. A P E S の効果

A P E S を使用してプログラムの設計を行うことで、工程の上流での設計品質を確保し、工程の後戻りを無くすことができ、プログラムの設計効率の向上を図ることができる。

図-9の左のグラフは、従来発生しているバグを原因別に分類し、その割合を示したものである。A P E S では、仕様書に記述されているモジュールを自動的に配置しているので、仕様の誤解・見落としによるモジュールの選択・配置ミス、R A

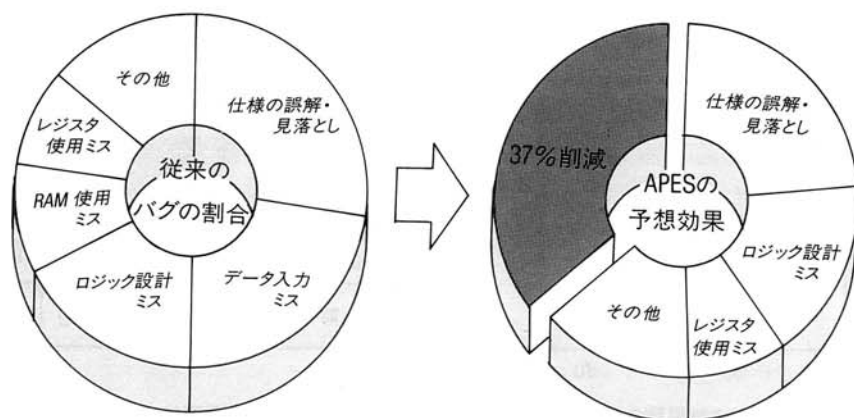


図-9 信頼性向上

Fig. 9 Enhancement of reliability

M使用ミスを防止できる。また、仕様書に記述されている制御定数を自動的にプログラムに書き込んでいるので、データの入力ミスによるバグを防止することが可能である。これらのことから、従来発生しているバグに対しては、図-9の右のグラフのように全体で37%の削減効果を予測できる。さらに、モジュールの標準化率が高まれば高まるほど、バグの削減効果は大きくなる。

また、図-10に示すように、従来のプログラミングに比べ、「モジュール配置」、「データ書込み」、「RAM割付け」、「共通部品配置」などの作業をAPESがサポートすることになり、手作業によるミスも防止できるので、プログラム開発工数の約30%の削減が期待できる。

5. あ と が き

本システムは、自動車用電子制御機器のアセンブル・プログラムを対象に、(1)プログラムの信頼

性の確保、(2)プログラムの設計工数の削減を図るために、仕様書の抽出データに基づいて標準化したプログラム部品を合成するという手法を用いて開発されたものである。この基本的な考え方は、一般の制御用アセンブル・プログラムへも応用が可能であると考えられる。

制御用プログラム言語として、アセンブル言語はまだ広く使われている。アセンブル言語は、限られたメモリに格納するためプログラミングテクニックにより、コンパクトに設計できる反面、大規模なプログラムを開発するには工数がかかり過ぎる欠点がある。このため、メモリに比較的余裕のあるシステムではC言語によるプログラムの開発も行われており、当社においても、プログラム開発のコスト削減のために、C言語によるプログラム開発を推進している。今後、本システムに用いられている構想や実現手法などのシステムの基本的な考え方を取り入れ、APESをC言語プロ

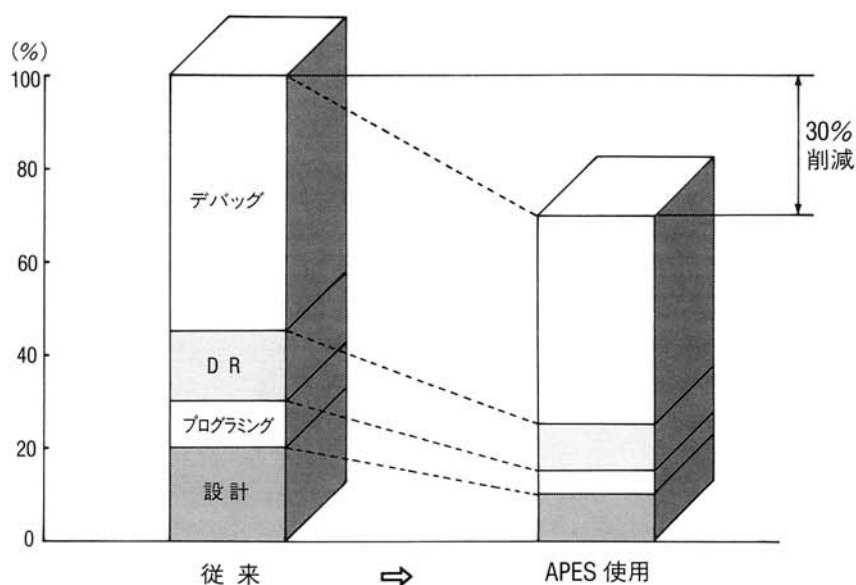


図-10 開発工数の削減

Fig.10 Decrease of developing cost

グラムにも適用できるシステムに発展させていきたい。

参考文献

- 1) 藤野: “CASE環境の概要”、情報処理、Vol.31、No.8、pp.1013-1019 (1990)
- 2) 原田: “事務処理ソフトウェアを対象としたCASEの現状と動向”、情報処理、Vol.31、No.8、pp.1036-1048 (1990)
- 3) 山本: “制御ソフトウェアを対象としたCASEの現状と動向”、情報処理、Vol.31、No.8、pp.1057-1067 (1990)
- 4) 技術士ソフトウェア研究会編: ソフトウェア生産工学ハンドブック、フジ・テクノシステム、pp.40-45 (1991)
- 5) 富士通編: システム開発標準 SDEM90 概説書、富士通、p.93 (1990)
- 6) 斗納、高橋、安木、内丸: “プログラム自動解析システム”、富士通テン技報、Vol.8、No.1、pp.55-62 (1990)
- 7) 井手、八木、斗納: “エンジンコントロール用自動デバッグシステム”、富士通テン技報、Vol.9、No.1、pp.69-78 (1991)