

アプリケーション開発・保守一貫支援システム (YPS/APG)
の適用による効果的な再利用方法

2S-5

廣瀬 守克

(富士通株式会社)

1. はじめに

ソフトウェア開発の効率化や品質の向上を阻害する要因としては、工程間での情報の書き換えの発生や、複数プログラム間での、共通情報の重複記述がある。

これらの問題を解決すべく、上流工程からのシステム開発の支援を目指した、アプリケーション開発・保守一貫支援システム、YPS/APG (YPS based Application Program Generation system)を開発した。(YPS/APGの概要については第38回情報処理学会全国大会で紹介済である)。

本稿ではYPS/APGの適用による、効果的な再利用方法について述べる。

2. YPS/APG 適用時のプログラム構成要素

YPS/APG では、プログラムを次の3つの要素に分けて考えている。

- (a) スケルトン : プログラムの骨格となる処理の流れを規定したもの(マッチング処理やグループトータルの処理など)で設計時の情報より自動生成される。
- (b) 部品 : プログラム間で共通な処理(ファイル読み込み処理やレコードの書き出し処理など)。
- (c) 固有処理 : そのプログラムに固有な処理。

YPS/APGはこの3つの要素を組み合わせることで1本の完成したプログラムを合成する。(図1)

合成結果はYPSで記述された仕様書の形式になっており、合成YPS仕様書と呼んでいる。

3. 設計情報の再利用

(1). データ項目情報の再利用

データ項目は、プログラムを構成する基本要素であり、同一のデータ項目は複数のプログラムで使用されることが多い。従って、同じ意味のデータ項目には、同じ名前をつけるなどの標準化を行い、ディクショナリで管理する。

定義する情報には、データ項目名や桁数、タイプなどデータ項目の属性情報の他に、表示属性、チェック仕様、サンプル値などの情報を定義する。

これらの情報は、レコード定義時に、レコードで使用されるデータ項目の、属性を定義する際にディクショナリを参照することにより再利用される。また、画面仕様定義時には、データ項目の属性情報のほかに、入力されたデータのチェック仕様についてもディクショナリより引用することにより再利用が図られる。

(2). 実イメージで定義された情報の再利用

YPS/APGは、画面や帳票のレイアウト、画面遷移図およびジョブフロー図については実イメージで定義する機能を有する。これらの定義情報は、従来のソース形式の定義体と比較して、現状の仕様が理解しやすいため、複写変更方式での再利用が行いやすい。

(3). ファイル仕様情報の再利用

ファイル仕様情報は、ファイルの基本的な属性やアクセスキー情報など、どのプログラムでも同一記述になるものと、アクセス名やオープンモードなど、個々のプログラム毎に異なる記述になるものを、分離して定義することで同一記述部分の再利用が可能となる。

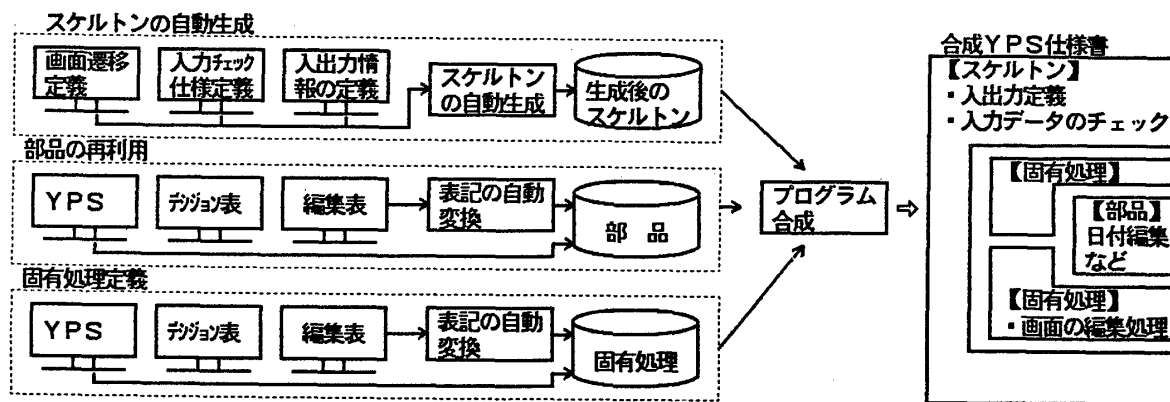


図1. YPS/APGにおけるプログラム合成 (オンラインプログラムの場合)

(4). スケルトン生成時に於ける設計情報の再利用

個々のプログラム毎のスケルトンは、基本となるスケルトンのひな型に、ファイルの仕様に関する情報とプログラムの入出力に関する情報を付加することにより生成される。オンラインプログラムの場合は、スケルトン生成時に画面遷移の定義情報や入力チェックの仕様についても引用される。

スケルトンのひな型、ファイルの仕様および入力チェックの仕様などの基本情報が再利用されている。

スケルトンのひな型は、オンライン用3本、バッチ用11本を用意している。また、独自のスケルトンのひな型を登録して再利用することも可能である。

4. 部品の再利用

プログラム間で共通な処理を『部品』として予め準備しておき、これらの再利用を考慮しながらシステム設計を行う。このようにすると、プログラム設計工程で新規に設計が必要となるのは、そのプログラムに固有な処理のみとなる。

部品の記述は、YPS表記のほか、デシジョン表や編集表での表記も可能である。

部品には、部品中で使用される各種名標（データ項目名、部品に組み込まれる下位の部品名など）を可変にし、部品を使用するときに自由に変更する機能がある。この機能により、手続き（ロジック）を名標に依存しない形でとりだし、部品として定義することを可能にしている。ここでは、再利用効果のある部品について説明する。

(1). サブルーチン呼出し型の部品

依頼パラメタのセット、サブルーチンの呼び出し（CALL命令）、復帰後のリターンコードのチェック処理の部分を取り出して部品として定義する。この部品を経由して、サブルーチンを呼ぶことで、インタフェースミスが減少する。この場合、可変要素をできるだけ少なくして単機能とする方向で作成する。

(2). 処理の制御を決定する部品

会話処理における一連の操作手順を部品として定義して、複数のプログラムで利用することで、操作の手順が標準化できる。例えば、入力チェック部品として入力項目の単独チェック処理の組み込み口と関連チェック処理の組み込み口を、この順に並べて部品を作成し、複数のプログラムで利用すれば、入力チェックの順序が簡単に標準化できる。

(3). ファイルの入出力部品

ファイルアクセスの記述は、レコード名やファイル名が異なる以外は、共通の記述となるため部品にしやすい。最適な入出力部品を指定すると、YPS/APGは、部品中で可変要素として定義してある仮のファイル名や仮のレコード名を、設計時の情報を参照して実際にプログラムで使用するファイル名やレコード名に、置き換えを行う。

その他に、異常処理ルーチン、エラーメッセージ出力後の応答処理、文字列の処理、出力電文の編集処理などの部品が再利用されやすい。

5. 再利用を促進するための機能

YPS/APGには、部品をキーワードによって検索する機能があるので、同種の部品をグループとして検索することができ検索効率を高めている。

6. 再利用の効果

(1). YPS仕様書記述（コーディング）量の減少

設計時の情報より仕様書記述を自動生成するため、特に、ファイル関連記述のような共通的な設計情報を中心にYPS仕様書の記述量が減少する。

今回の社内試用時の実績として、固有処理としてどれだけ記述する必要があったかを、生成後の全プログラムステップ数と記述した固有処理ステップ数との比で表してみると、

オンラインの場合の固有処理記述率 = 42%
バッチの場合の固有処理記述率 = 43%
という結果が出ている。

このとき

$$\text{固有処理記述率} = \frac{\text{固有処理ステップ数}}{\text{生成後プログラムステップ数}}$$

である。

固有処理として作成しなければならなかった処理は従来に比べ半減している。

(2). テストレス指向

生成されたスケルトンと予め準備された部品については、品質が確認されており、原則としてテストが省略できる。また、固有処理を作成する場合でも、デシジョン表や編集表のように表形式で記述できるため机上レビューでのチェックが可能となり、単体テストの負荷が軽減できる。

7. むすび

以上、YPS/APGの適用による再利用方法について述べた。今後は、大規模システム開発に適用し、再利用の効果を評価したいと考えている。また、適用事例を分析し、メーカー提供の標準部品の充実を図る予定である。

【参考文献】

- 1) 杉田他：アプリケーション開発、保守一貫支援システム -YPS/APG-、第38回情報処理学会全国大会
- 2) 富澤他：部品化指向による生産性向上への取り組み 富士通ジャーナルNo.159