

保守ドキュメント自動生成システム「SysSurf」

— ドキュメント化手法 —

The document generating system "SysSurf"

- The explanation of generated documents for maintenance -

五味 俊明 上村 学 吉野 利明 松尾 昭彦 伊藤 栄信
Toshiaki Gomi Manabu Kamimura Toshiaki Yoshino Akihiko Matsuo Hidenobu Ito

(株) 富士通研究所 FUJITSU LABORATORIES LTD.

1. まえがき

世の中の様々な場面で、コンピュータシステムの資産が稼働している。これらを保守しようとした場合に、古い資産などでは資料がない、あるいは仕様の変更が重なり現在の資料がない、また資料はあるが概要などの理解に役立つものがないという状況があり、保守に取り掛かる際のシステム理解に手間取ることがある。

このような中で、筆者らは保守などでの概要理解を支援するドキュメント自動生成システムとして、SysSurfを開発した。背景などは[1]を参照されたい。本稿では、SysSurfにより自動生成されるドキュメントについて紹介する。

2. SysSurf ドキュメント生成概要

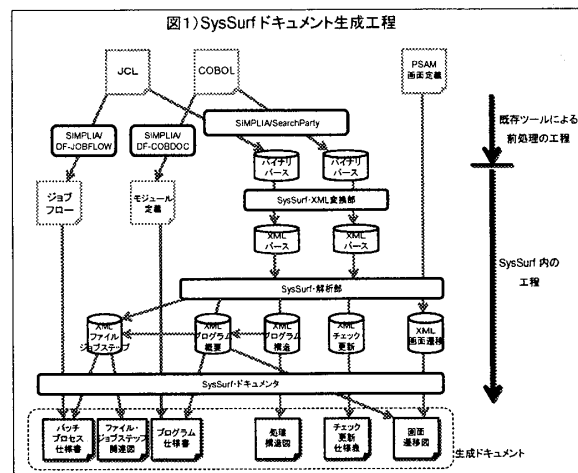
SysSurf で取り扱う資産は基本的にメインフレーム用 COBOL85 言語で記述されたものである。COBOL のソースプログラムと JCL というプログラムの起動制御が記述されたファイルと PSAM 画面定義体という画面定義を記述したファイルを対象としている。これらのソースコードから以下の6種類のドキュメントを生成している。

- プログラム内部理解を支援するためのドキュメント
 - 処理構造図：セクション呼び出し構造と対応する入出力情報を合わせて示した表形式の図。
 - チェック更新仕様表：変数への代入と代入が行われる条件、また変数が条件文で参照されるときに条件記述を示した表形式の図。
- プログラム概要理解を支援するためのドキュメント
 - プログラム仕様書：プログラムでのデータベースアクセスを含む入出力情報と主要セクションを記述した図。
 - 画面遷移図：画面の遷移とその条件、画面についての主な処理の列挙と入出力情報を示した図。
- プロセス概要の把握を支援するためのドキュメント
 - バッチプロセス仕様書：ジョブ（一つの JCL の実行）について全体としての入出力となっているファイル一覧とそれを参照、更新するプログラムを列挙して示した図。
 - ファイル・ジョブステップ関連図：ジョブについて、ファイルの入出力、生成から削除までをプログラムの実行順と対応させて示した表。

生成工程の流れを図1に示している。

SysSurf では、独自に生成しているドキュメントと、既存ツールで生成されるドキュメントに追加情報を加える形で用意するドキュメントがある。

生成の流れは、まず前処理工程として、JCL,COBOL を影響検索ツール「SIMPLIA/SearchParty」(*1)でパースする。また、既存ツール「SIMPLIA/DF-COBDOC」(*2)、「SIMPLIA/DF-JOBFLOW」(*3)で、モジュール定義図とジョブフロー図を生成する。次にパースした情報から XML 変換部で XML の論理構造に変換する。これはソースコードとほぼ同等の情報である。それをさらに解析部で加工を加え、各ドキュメントに出力するための XML 情報に変換する。PSAM 画面定義体も解析部で変換する。これらの変換して用意した XML 元情報からドキュメントにて6種類の表形式またはワードプロセッサの文書としてドキュメントを生成する。ドキュメントの処理には、コメント継承の処理も含まれている。



3. 各ドキュメントについて

3.1 処理構造図

COBOL ソースの例と対応する処理構造図を図2に示す。COBOL ではサブルーチンをセクションと呼び、従来からもセクション構造図(*4)で、プログラムの構造は示されている。しかし、入出力など重要な処理が行われるセクションの目星がつけにくいという状況があった。

処理構造図では、セクションの構造と共に、入出力が行われるセクションについてはその情報も対応する場所に掲載している。また、CALL 文で別のプログラム（サブプログラム）を呼び出している場合は、セクション構造の途中に CALL 文の欄を挿入する。その内部で行われる入出力についても掲載する。さらにその内部で呼ばれるサブプロ

(*1) COBOL 向け影響検索ツール、富士通製

(*2) COBOL 向けモジュール定義図等の生成ツール、富士通製

(*3) JCL 向けジョブフロー図等の生成ツール、富士通製

(*4)(*2)で生成されるドキュメントの一つ

ラムについても追跡し、呼び出しがなくなるまで追跡して入出力を取得して掲載している。

セクション構造図では、セクションが2回目以降登場する場合は、省略されることがあるが、処理構造図では基本的に記述のあるまま順番通り表示する。これにより、プログラム構造のボリューム感、複雑さについても把握することができる。

各呼び出しの欄にその実行条件を挿入できる。あるセクションで呼び出しが行われる場合、そのセクション内で、呼び出しにかかる条件すべてが列挙される。また、EVALUATE - WHEN 文は IF 文での条件記述に展開して記述しており、ループも条件の一つとして記述する。

これらの情報からプログラム構造の概観を掴むことができる。

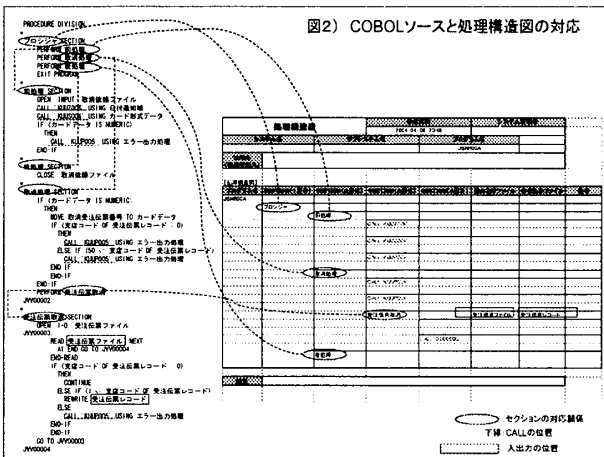


図3) チェック更新仕様書の例

更新種別	更新対象	更新条件	更新内容	更新後
新規	システム	システム	システム	システム
変更	システム	システム	システム	システム
削除	システム	システム	システム	システム

3.2 チェック更新仕様表

チェック更新仕様書の例を図3に示す。ここでは、ファイルアクセスで利用される変数について、更新情報(変数に代入を行うときの代入元)と更新が行われるときの条件、あるいは変数がチェックされる(条件文で参照される)ときの条件式について列挙している。更新の条件はプログラム内で遡ることができる。また対象にする変数の指定や、条件の追跡範囲の指定ができる。

今回、注目する変数と範囲を限定することで、実用的なサイズの表を提供している。

3.3 プログラム仕様書

ここではモジュール定義図に新たにデータベースの入出力情報、呼び出し先プログラム内部も含めた入出力情報、主要なセクションを追加している。

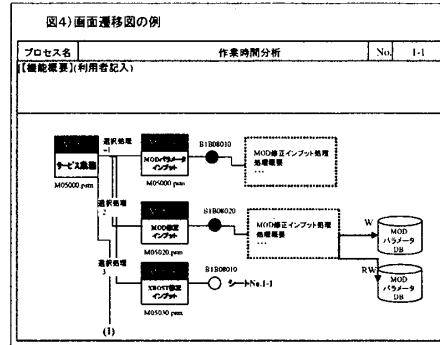
主要なセクションとは、処理構造図で記述されたセクションのうち、階層の2層目と3層目に来るものである。特にセクション名が日本語になっていればこの情報は参考になる。このように、従来のものに対してより詳細な入出力情報と、主要な内部構造情報を提供している。

3.4 画面遷移図

画面遷移図の例を図4に示す。これはPSAM画面定義と画面から呼ばれるプログラムの情報から構成される。

ここでは、画面、プログラム、入出力情報を遷移線で結んでいる。プログラム仕様書の入出力情報と主要なセクションの情報に相当するものが記述される。

この図を通じて、画面の遷移を把握することができる。



3.5 バッチプロセス仕様書

これはジョブフロー図にジョブ全体についての入出力ファイルと主要なプログラムを追加したものになっている。主要なプログラムとは、全体の入出力になるファイルにアクセスするプログラムである。ここではジョブフロー図だけでは特定できない入力または出力の判断が、プログラムの情報を利用して可能となっている。

3.6 ファイル・ジョブステップ関連図

これはジョブの中の各ジョブステップ(プログラムの実行)とファイルとの関係をマトリックス形式にまとめたものである。各ステップが読み書きするファイルの位置に読み書きと生成、削除の状態が示される。読み書きについてはプログラムの情報から判断している。最後の行にはジョブ全体として、そのファイルが入力になっているのか、出力になっているのかを表示する。

ここでは、全体の入出力となるデータを容易に知ることができ、ジョブとデータの関係が把握できる。

4. まとめ

これまでいくつかのプロジェクトの現場のSEに SysSurf を試用いただき、以下の評価をいただいた。

- 処理構造図、バッチプロセス仕様書：概要理解に有効
 - プログラム仕様書：仕様書整備に有効
 - ファイル・ジョブステップ関連図：ジョブステップ間のファイルの生存期間を把握するのに有効
 - 画面遷移図：業務レベルの情報が必要
 - チェック更新仕様表：設計時のものとの比較に有効
- まだ概要理解に必要な情報が不足しており、改善を検討する予定である。

参考文献

[1]上村学他：保守ドキュメント自動生成システム「SysSurf」による保守支援 FIT2004 (2004)
 [2]富士通：SIMPLIA/DF-COBDOC Document Guide
 [3]西村 恕彦、植村 俊亮：入門 COBOL(新版) オーム社(1993)
 [4]富士通：OS IV/MSP ジョブ制御言語文法書 AF II V10用(1996)
 [5]富士通：SIMPLIA/DF-JOBFLOW Document Guide