

CAPSDFにおけるリエンジニアリング

津田道夫

(株)日立製作所

ビジネスシステム開発センタ

最近の経済状況等を背景に、既存ソフトウェア資産の保守性向上を図る、標準化/構造化により再活用を図るリエンジニアリングに期待が出ている。日立では情報システムの開発・保守を支援するアプリケーション開発支援体系CAPSDFの一環としてリエンジニアリング技術を開発し、事業化した。支援ツールは製品化して提供する他、この技術を用いたリエンジニアリングサービスも提供する。本論文ではサービスの視点から、資産分析診断、プログラム評価、ソフトウェア標準化機能を述べ、特に技術的特徴である同値解析について述べる。同値解析とはプログラムのデータ名称の異音同義語を抽出する機能で、修正変更波及解析の作業で用いられる。

Reengineering of CAPSDF

Michio Tsuda

Institute of Advanced Business Systems

Hitachi Ltd.

Given the recent difficult economic situation, software reengineering, i.e. increased maintainability of existing software and the attempt of practical reuse via standardization and/or structure, is being anticipated. At Hitachi Ltd., we have developed a reengineering technology as a link in the chain of CAPSDF, an application development system which supports the development and maintenance of information systems. Hitachi supplies both the support tool product as well as software reengineering service. In this thesis, we will describe an assets analysis and diagnosis, a program evaluation and software standardization function. We will also describe Equivalency Analysis, which is a special feature of this software reengineering technology. Equivalency Analysis is a function that extracts equivalent meaning words with different pronunciation from data names in programs, and is used in work of Modify Propagate Analysis.

1 まえがき

日立アプリケーション開発支援体系CAPSDF(Computer-aided Application System Development Framework)は、情報システム計画技法／開発方法論と統合型CASE(SEWB3: Software Engineering Work Bench3)からなっている。

本論文では、CAPSDFの支援ツールであるリエンジニアリング支援ツールを用いたサービス事業について、その仕様と実用化技術及び今後の課題を述べる。リエンジニアリングサービスは資産分析診断、ドキュメント作成、ソフトウェア標準化、CASE移行等7種類のサービスがある。対象とするソフトウェアは古いプログラムのため、ドキュメントがない、開発担当者がいない等により、保守作業に支障を与えている場合が多い。またプログラムが標準化されていないので、データ名称に同音異義語、異音同義語が存在している、永年の修正作業によりプログラム間に矛盾や不整合が発生している。

支援ツールはこれらのソフトウェアを解析し、問題点を抽出する。例えば同値解析機能ではJCL(Job Control Language)やソースプログラム(COBOL)の情報から異音同義語を検出する。もとより設計理由や意図がソフトウェアには表現されていないので支援ツールでの完全な自動検出は不可能である。支援ツールが検出した問題点をリエンジニアリングエンジニアが評価して、標準化作業を進めるプロセスになる。ここにリエンジニアリングサービスの難しさがある。

2 リエンジニアリングの概要

2.1 背景と目的

大規模な新規開発が一巡し、保守作業の効率化に関心が集まっている。またこれからの情報システム開発を考察すると、基幹業務は新規開発型から既存ソフトウェア流用型の構築へ、ダウンサイジングによりホスト業務からCSS(Client Server System)業務の再構築へ移りつつある。ま

た新規開発の場合でも、既存ソフトウェアから設計情報を抽出して再利用したいとのニーズもリエンジニアリング技術開発の背景にある。

フォワードエンジニアリングの支援ツールにも、保守作業を支援するツールが従来より多く開発されて来た。しかし対象とするソフトウェアは標準化されており、それをベースに保守と再利用を実現している。例えばリポジトリ格納情報によるクロスリファレンスやインバクトレポートの作成、コーディングルールによるプログラムドキュメントの作成、日本語木構造チャートの作成がある。またデータ中心型開発(DOA: Data Oriented Approach)では標準データの名称と操作をカプセル化して、保守作業の局析化を図っている。

SEWB3では上記の保守支援機能をサポートしているほか、設計仕様書とリンクを張った設計メモが記述できる。設計者は設計作業中に、設計理由やアイデア等を自由に記録として残せる。

リエンジニアリングが対象とするソフトウェアは標準化されておらず次のような問題点を持っている。

- ・ソフトウェアの資産管理がされていない。
- ・設計仕様書等のドキュメントが不正確。
- ・変更の繰り返しによりプログラムが複雑化したり、矛盾点を持っている。
- ・プログラムの変更部分の特定化が困難。
- ・古いバージョンの言語仕様を使用している。
- ・アセンブラ言語や簡易言語の混在。

日立大型システムのユーザでは、43%が1Mstep以上のプログラムを保守している。特にDB使用ユーザでは平均1.4Mstepを保有している。

リエンジニアリングの目的は保守コスト削減と既存ソフトウェア資産の再利用による開発コストの削減であり、次に示す作業を実現する。

- ・ソフトウェアの分析、診断、整理により資産の棚卸と管理を標準化する。
- ・設計情報を抽出して、上流工程での保守作業

を実現する。

- ・ソフトウェア解析によりプログラム変更範囲を特定化する。
- ・ソフトウェアを標準化して、フォワードエンジニアリングの保守支援機能を活用する。

2.2 リエンジニアリングの体系

リエンジニアリングとは「既存ソフトウェアの標準化／構造化により、ソフトウェア保守と再利用を支援する技術」である。しかし一般的な定義はまだなく、その分類体系も確立していない。中部大学の竹下は「リエンジニアリングは、リストラクチャリング(再構造化)、リバースエンジニアリング(逆エンジニアリング)、およびその結果行われるフォワードエンジニアリングによるシステム再生成(regeneration)を包含する統括的技術(umbrella technology)」と定義している。

本論文では、既存ソフトウェアを直接解析して、構造化／標準化するまでをリエンジニアリングの範囲とする。図1にリエンジニアリングの概念を示す。

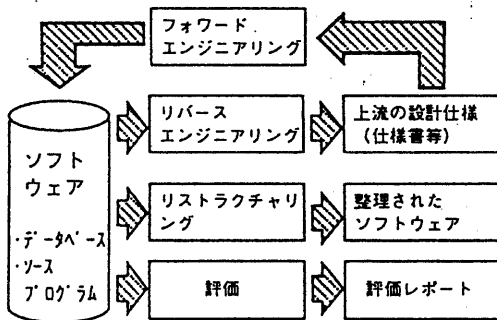


図1 リエンジニアリングの概念

リエンジニアリングは「リバースエンジニアリング」、「リストラクチャリング」、「評価」の3種類に大別される。

(1) リバースエンジニアリング

既存ソフトウェアを解析して上位の設計仕様を抽出して、現行仕様を理解する。図2にリバース

エンジニアリングの詳細を示す。実用化のレベルではまだ下流工程でのリバースが多い。具体的には次に示す設計仕様のリバースを実現している。

- ・ジョブフロー
- ・プログラム設計ドキュメント
- ・日本語木構造チャート(PAD)
- ・変更波及解析用クロスリファレンス
- ・設計仕様(ファイル、レコード、書式)

(2) リストラクチャリング

ソースプログラムを論理ロジックを変更することなく、表現形式を標準化する。プログラムを読み易くするための清書機能とデータ名称を統一するデータ名称標準化支援機能がある。

(3) 評価

ルールとして登録されているコーディング基準(約80種類)に基づいて、プログラムを評価する。

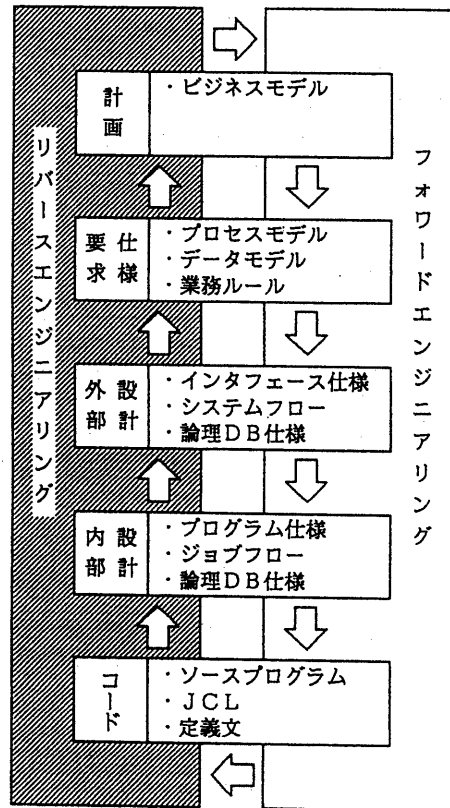


図2 リバースエンジニアリング

3 リエンジニアリングサービス

日本のコンピュータ産業は事業構造を変革しようとしている。ハードウェアからソフトウェア主体へ、基本ソフトウェアからコンピュータサービスの提供に移りつつある。

日立ではこのサービスをソリューションビジネスと名付けて、サービス体系FOREFRONT SS (Solution Service)として提供している。リエンジニアリングサービスもその一環として位置付けられている。図3にサービスの流れを示す。

(1) 資産分析診断

JCL, ソース, ロード等の資産調査を行い、一覧表とサマリ表から実態を把握する。次に関連情報と未対応チェックリストを作成して、未使用/不足ソフトウェアを検出する。これらの情報に仕様書の整備度等を加味して資産を分析して、診断報告書を作成する。分析結果は各種リエンジニアリングサービスの実施計画に反映する。

(2) 資産整理

資産分析で作成した診断報告書に基づき、ユーザは資産の取捨選択の方針を立て、資産を整理する。

(3) サンプルリエンジニアリング

既存ソフトウェア資産からJCL, ソース等を数十本程度サンプリングして、サービスを試行する。この作業によりリエンジニアリングサービスの効果と概算費用を見積る。

(4) リドキュメント

JCL, ソースプログラムから設計ドキュメントを作成する。プログラム処理の流れを示すジョブフロー図や各種設計仕様書(ファイル, レコード, 書式)を作成する。

(5) プログラム評価

コーディング基準に基づいて、プログラムの記述内容を評価する。

(6) ソフトウェア標準化

データ名称の標準化とプログラムの標準化を行なう。

- ・ファイル/レコード/データ項目名称を解析して、データ名称を標準化する。
- ・プログラム内のデータ名称を標準名称に置換して、標準化する。
- ・プログラムのレコード記述を共通のCOPY原始文に変換する。

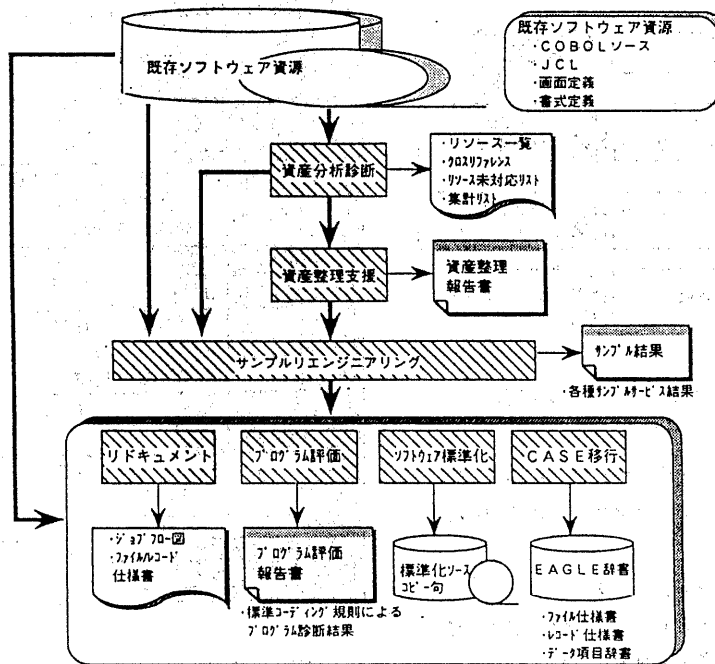


図3 リエンジニアリングサービスの流れ

(7) CASE移行

統合型CASE EAGLE2/SEWB3のフォワードエンジニアリング環境に移行する。標準化した設計仕様をライブラリやリポジトリに登録する。

- ・ファイル/レコード仕様書の登録
- ・標準データ項目の辞書登録
- ・COPY原始文から作成したレコード仕様書の登録

以上リエンジニアリングサービスについて述べたが、本サービスの中心となるソフトウェア標準化技術について次に述べる。

4. ソフトウェアの標準化

CAPSDFではソフトウェア開発技法としてデータ中心型開発技法(DOA)を中心に展開している。この技法は、データ項目を先ず正規化/標準化してリポジトリのデータ項目辞書に登録して、標準データの維持プロセスを一貫して設計する技法である。データ項目辞書にデータ名称、属性と操作(現在はチェック及び編集サポート)をカプセル化して登録する。

リエンジニアリングではソフトウェアを標準化して、DOAによる保守と開発を実現する。

ソフトウェアの標準化の流れを図4に示す。標準化は「データの標準化」と「プログラム」の標準化

に分けられる。

4.1 データの標準化

プログラムで用いられているデータの名称や属性を統一化して、更に基本的な共通項目であるドメイン属性や名称付与基準を設けて、名称からデータの目的・意味・カテゴリ等を類推すると保守や仕様理解が容易になる。

しかしながら既存ソースプログラムの多くは、名称が標準化されていないか、不十分な基準により作成されているので保守効率を下げている。そのためデータ名称の意味把握が困難、データ名称が無秩序に変化している、プログラム変更箇所の特が困難等の問題が発生している。また同じ意味のデータを異なる名称で用いていると、各々の関連性を把握するためには、データに関する処理を厳密に調査する必要がある。

図4のソフトウェアの標準化に示した作業の流れに従ってデータの標準化作業を説明する。

(1) 解析

リエンジニアリングはシステム、サブシステム、ジョブといった複数のソフトウェア群が対象になる。まず最初にこれらのソフトウェア(プログラム、COPY原始文、JCL)を解析して、確認用ドキュメントと矛盾点や不整合性を抽出した未対応リストを出力する。未対応リストはロード/ソース未対応リスト、データセット/ファイル未

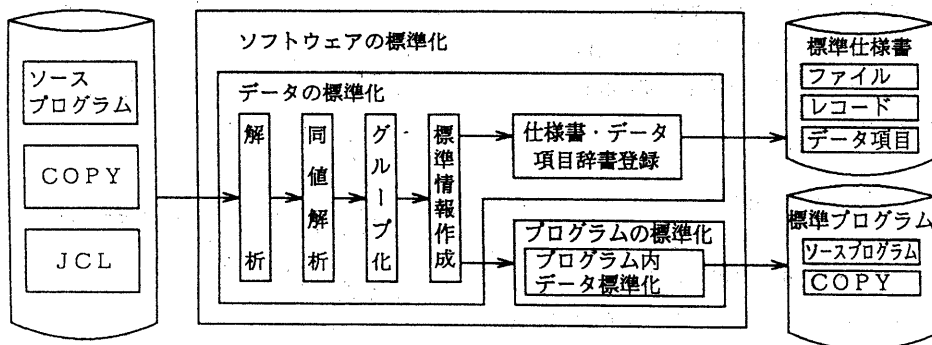


図4 ソフトウェアの標準化

対応リスト等がある。例えばJCLのロードモジュール名とライブラリのソースプログラム名が不一致の場合、後で述べるデータ項目間の関連を抽出する同値解析とグループ化が出来なくなるので、作業者が対応関係を入力しなければならない。

(2) 同値解析

不統一のデータには、同音異義語(ホノニム)と異音同義語(シノニム)がある。同値解析では、このうち異音同義語を同値データとして抽出する。

同値データは、物理的な同一エリア使用による同値と論理的な結び付けによる同値の2種類がある。前者はJCLとソースプログラムの関連から抽出し、後者はソースプログラムのデータ間の関連から抽出する。

(3) グループ化

同値解析ではデータの2者間情報しか抽出しないので、同値データから同値グループを作る。同値グループはファイル、レコード、データ項目の3種類ある。

グループ化では同値データから下位波及分析を行なう。下位波及分析とは同値データの下位のレベルの同値解析をすることである。例えば同値関係のあるレコードは、それを構成するデータ項目の同値解析を行なう。

作業者はグループ化の結果を検証して、削除と追加の編集を行なう。

(4) 標準情報作成

グループ化したデータに対し、標準名称と属性を付与して標準データを作成する。

(5) 仕様書・データ項目辞書登録

フォワードエンジニアリング用の仕様書(ファイル、レコード)、データ項目辞書に標準データを登録する。

4.2 プログラムの標準化

既存ソースプログラム、COPY原始文を、ファイル、レコード、データ項目の標準化データに置換する。図5はプログラムの標準化例である。標準化前のプログラムはデータ名称からは何をしようとしているのか、処理ロジックが理解できない。

また名称が不統一なのでプログラム間の関連も不明で、業務仕様の変更に伴うプログラム修正箇所の特定化も困難である。

図5では2本のプログラムを解析して、同値データを抽出している。この場合ファイル「KANRI-FILE」と「IN-FILE」が同値であった。次にファイルを構成するレコード、データ項目への下位波及分析をすると「KANRI-INF」と「IN-DATA」が同値関係にあり、更に下位波及分析の結果「CODE-NO」と「SYOHIN-ID」が同値であると判明した。またソースプログラム1のロジックを解析して、「CODE1」と「CODE-NO」の同値関係を抽出した。

抽出した同値データと標準データにより、ソースプログラムを置換する。置換した例が図5の下段のプログラムで、保守作業者にとって理解しやすい表現形式になっている。この例では、2つの

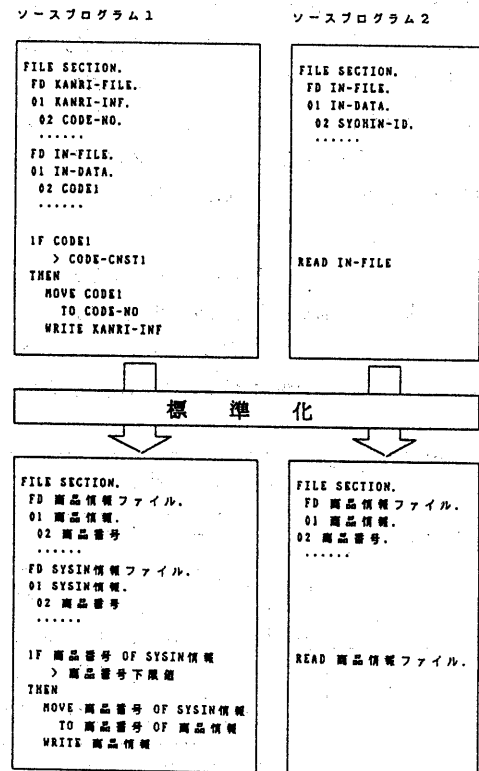


図5 プログラムの標準化

プログラムが同じ「商品情報ファイル」を使用して
おり、プログラム1は「SYSIN情報」をチェ
ックして「商品情報ファイル」を作成している。プロ
グラム2はこの「商品情報ファイル」を入力してい
ることが容易に分かる。

5 同値解析

プログラム間、及びプログラム内の異音同義語
(シノニム)を同値データと呼ぶ。「同値」は、物理
的実体が同一である場合(物理的同値)か、もしく
は論理的な結び付けが等価の場合(論理的同値)に、
同値関係が成立する。「同値」対象データは、ソー
スプログラムで記述されているファイル名である
「論理ファイル名」、01レベルで記述されている
「レコード名」、02レベル以降のデータ名である
「データ項目」の3種類である。

同値解析の目的は、主として次の3つである。

- ・ソースプログラム内のデータ名の物理的な関
連と論理的な関連を抽出し、プログラム修正
作業における影響範囲を明示する。更にグル
ープ化の機能によりサブシステムやジョブと
いった単位のプログラム群での変更波及解析

を容易にする。

- ・同値解析とグループ化によりデータの標準化
を支援する。標準データはデータ項目辞書に
登録され、開発と保守に利用される。
- ・プログラムを標準化して、意味理解を容易に
する。

同値解析では、同値情報としてサブシステムの
名称、同値種別、同値元(比較元)情報、同値先
(比較先)情報を抽出する。同値元データとはデー
タ転送元や再定義前のデータで、同値先データと
は転送先や再定義後のデータである。同値元と同
値先を識別して、データの流れの方向性を把握し
て変更波及分析に利用する。

(1) 物理的同値

物理的同値とは、同一物理エリアをデータが共
有している場合の同値である。「ファイル」の同値
解析では、2つのプログラムにおいてJCLのD
SN名(データセットの固有名称)が一致する場合
や、同一ロードモジュール内のプログラムでDD
名(プログラムの論理ファイルとJCLのデー
タセットを関連付けるデータ定義名称)が一致する
場合を同値としている。

図6はDSN名が一致した時の同値例である。

```

『          <ジョブステップ1>          . . .          『          <ジョブステップ2>
// JOB P1  JOB
// STEP EXEC PRG=P1
// FILE11 DD DSN=DSN11, . . .

          <ソースプログラムP1>          . . .          <ソースプログラムP2>
IDENTIFICATION DIVISION.
PROGRAM-ID P1.
ENVIRONMENT DIVISION.
INPUT-OUTPUT SECTION.
FILE CONTROL.
SELECT *FILE11
ASSIGN TO FILE11.
DATA DIVISION.
FILE SECTION.
FD 771101 . . .
01 771101.
05 771101

:
』          :          』

```

図6 同一DSNによる「ファイル」同値解析例

「ファイル名01」と「ファイル名02」は、DSN名「DSN11」が一致するのでこれらのファイルは同値関係にある。

「レコード」と「データ項目」は、主に再定義の場合、同値としている。

(2) 論理的同値

論理的な同値とは、ソースプログラム内やプログラム間のデータ転送等による同値である。MOVE文やREAD文、WRITE文等を解析して同値解析をする。プログラム間の同値データはCALL文のデータ並びの解析から同値解析する。図7はCALL文の同値解析の例である。データ「A」は「D」と、「B」は「E」と同値関係がある。

```
PROGRAM-ID. S1.  
FILE SECTION.  
FD FILE01.  
  01 A .....  
WORKING-STORAGE.....  
  01 B .....  
  01 C .....  
CALL 'S2' USING A B.
```

```
PROGRAM-ID. S2.  
PROCEDURE DIVISION USING D E.  
  --  
  --
```

図7 CALL文の同値解析

6 おわりに

本論文ではCAPSDFにおけるリエンジニアリング技術を、サービスの観点から述べた。しかしまだ技術課題は多い。ソフトウェアの標準化を例にしても、システム全体の同値解析の方式、同値抽出のルールベース化とカスタマイズ化、効率的なグループ化方式等の課題が残っている。また標準化後のシステムの検証と評価方式を確立する必要がある。

リエンジニアリングは、データとプロセスを対象とした技術が実用化されているが、ルールを対象とした研究も行なわれている。プログラムの意味理解による操作部品の抽出技術やデータ中心分析によるプログラム論理の抽出技術がある。

リエンジニアリングに対するユーザのニーズは多様である。今後はニーズとリエンジニアリング技術の実用化動向を踏まえてサービスの拡充に反映させて行きたい。

参考文献

- [1] 日本情報処理開発協会：CASEに関する調査研究報告書，PP.135-137，1991.3.
- [2] 原田，他：CASEのすべて，オーム社，PP.315-328，1991.11.
- [3] 渡部，他：ソフトウェア資源の再利用を目的としたリエンジニアリング支援ツールの開発と適用，日科技連第12回ソフトウェア生産における品質管理ランボジウム，1992.9.
- [4] 佃，他：CSS統合開発環境(7)-リエンジニアリング-，情報処理学会第45回全国大会，PP.5-351，1992.10.
- [5] 岡部，他：リエンジニアリング支援ツールの適用と評価，情報処理学会第46回全国大会，PP.5-231，1993.3.
- [6] 津田：CAPSDFにおけるリエンジニアリング，情報処理学会研究報告93-IS-42，PP.77-85，1993.1.
- [7] 山川，他：データ中心分析によるプログラム論理の抽出，情報処理学会研究報告93-IS-42，PP.107-115，1993.1.
- [8] 竹下：ソフトウェアの保守・再開発と再利用，共立出版，1992.8.