情報システム 42－8
（1993. 1. 19）

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

情報システムの開発・保守を支援する日立アプリケーション開発支援体系ＣＡＰＳＤＦの一環として、既存ソフトウェアの再標準化／再構造化を図るリエンジアリング支援ツールを開発した。支援ツールは既存ソフトウェアから上流の仕様情報を抽出するリバースエンジニアリングツールと既存ソフトウェアを再標準化するリストラクチャリングツールからなるが、本論文ではこれら支援ツールのうち、ジョブフロー逆生成ツールとデータ名称標準化ツールの機能と評価について述べる。リエンジニアリング支援ツールはフォワードエンジニアリング開発支援システムＳＥＷＢ／ＥＡＧＬＥと連携してソフトウェアライフサイクルの一貫支援を実現している。

Re-engineering in CAPSDF

Michio Tsuda
Institute Of Advanced Business Systems, Hitachi, Ltd.
Hitachi Systemplaza Shin Kawasaki 890 Kashimada,
Saiwai, Kawasaki, 211, Japan

We have developed "re-engineering" support tools that realize restandardization and/or restructure of existing software, as a link in the chain of HITACHI's CAPSDF, an application system, which supports the development and maintenance of information systems. These support tools consist of a reverse engineering tool, which extracts up-stream information from existing software, and a restructuring tool, which re-standardizes existing software. In this thesis, I describe the functions and merit of the generation job-flow chart tool found in these support tools. These "re-engineering" tools provide support all the way through the software lifecycle, via connection to SEWB/EAGLE, the forward engineering support system.

１．まえがき

ＣＡＰＳＤＦ（Computer-aided Application System Development Framework）は、情報システムのライフサイクルを一貫して支援する、日立アプリケーション開発支援体系で情報システム計画技法／開発方法論と統合型ＣＡＳＥ支援ツールからなっている。

本論文では、支援ツールのひとつであるリエンジニアリング支援ツールの機能と評価を述べる。支援ツールは既存ソフトウェアから仕様情報を抽出し、保守ドキュメント、設計仕様書を生成するリバースエンジニアリングツールと、保守作業や再利用を容易にする目的で既存ソフトウェアを再構造化／再標準化するリストラクチャリングツールがある。支援ツールのうちジョブフロー逆生成ツールと、データ名称標準化ツールの機能を述べる。対象とするソフトウェアは開発が古く標準化がされていない、永年の変更修正作業により、ソフトウェア相互間に不整合を生じているなどの場合が多く、支援ツールでは関連するソフトウェアを解析して仕様を抽出する機能が必要となる。

例えば、ジョブフロー逆生成ツールではＪＣＬ（Job Control Language）とソースプログラム（COBOL）を解析しジョブフローを生成するが、ＪＣＬだけで生成する場合は評価モデルの例では、ジョブフロー確定率が５１％であった。

支援ツールはまず既存ソフトウェアから必要な情報を抽出し、リエンジニアリングＤＢに格納する。次に関連情報を解析しリソース情報一覧表等のドキュメントと変更影響分析用のクロスリファレンスを出力するが、この時にデータ名称の同値解析をする。同値解析ではソースプログラムの解析を中心にファイル、レコード、データ項目の異名同値語（異音同義語）を検出し、同値情報としてグループ化する。ソースプログラムをこの同値情報により、標準名称に置換する。また、支援ツールはフォワード開発支援システムと連携するため、ＥＡＧＬＥ／ＳＥＷＢの仕様情報を生成する。

リエンジニアリング支援ツールにより、ジョブフロー生成作業で７倍、データ名称標準化作業で２．５倍の作業効率が向上する評価結果を得た。

２．ＣＡＰＳＤＦの体系

情報システムは、戦略的業務改革、ダウンサイジング、広域複合システム等多様なニーズへの対応が要求されている。ＣＡＰＳＤＦは、ソフトウェア開発の方法論／技法と統合型ＣＡＳＥ支援ツールの体系であるが、開発作業だけでなく膨大な既存システムの変更、拡張を効率的に行う保守作業支援体系も提供している。図１にＣＡＰＳＤＦの体系を示す。

情報システム計画技法　ＨＩＰＬＡＮ

企業レベルの戦略計画から情報システム全体計画、個別業務システム開発計画まで、レベルに対応した計画技法。

〔Hitachi Integrated Planning Procedure for Information Systems〕

情報システム開発方法論　ＨＩＰＡＣＥ

ＨＩＰＬＡＮで計画した情報システムを実現する開発方法論。開発標準手順と開発技法がある。開発技法としてデータ中心型開発技法、プロトタイピング技法、部品再利用によるプログラム開発技法、プロジェクト開発技法がある。

〔Hitachi Phased Approach for High Productive Computer System's Engineering〕

ＷＳ型開発支援ツール　ＳＥＷＢ３

ＵＮＩＸ　ＷＳ（製品名：３０５０）で稼動するクライアント／サーバ型統合開発支援ツール。ＣＡＳＥプラットフォーム（リポジトリを含む）とシステム設計者用ＳＥＷＢ３（５種類）、プログラマ用ＳＥＷＢ３（７種類）、プロジェクト管理者用ＳＥＷＢ３で構成する。

〔Software Engineering Workbench〕

ホスト型開発支援ツール　ＥＡＧＬＥ２

ホストコンピュータ（製品名：日立Ｍシリーズ）で稼動する統合開発支援ツール。

〔Effective Approach to Achieving High Level Software Productivity〕

第四世代言語　ＥＡＧＬＥ／４ＧＬ

ＵＮＩＸ　ＷＳ及び、小型コンピュータで動作する第四世代言語。大型ホストコンピュータからＷＳまで、共通の言語仕様で実行するターゲットフリー を実現している。

リエンジニアリング支援ツール

既存ソフトウェアを解析し、設計情報の抽出、標準化を行う。

リポジトリ

ソフトウェア資源を一元的に管理する。ＷＳリポジトリとホストリポジトリがあり、設計情報、保守情報、管理情報やデータ項目辞書、プログラム部品等の再利用情報も管理する。

ＳＥＷＢ３は、リポジトリをサーバに搭載し、クライアント（ＷＳ又はＸ端末）で各種開発支援ツールを動作させる分散開発環境で、ホストコンピュータからＷＳまでの業務システム（開発対象言語：COBOL,PL/I,C）を同一環境で開発する。

ＣＡＳＥプラットフォームは、ＥＣＭＡとＮＩＳＴのリファレンスモデル（トースタモデル）に基づきデータ統合、ツール統合、ユーザインタフェース統合、プロセス統合のメカニズムを提供している。ユーザインタフェースは、ＯＳＦ／Ｍｏｔｉｆで統合している。また、プロセス統合として開発作業をナビゲーションするメカニズムを提供しており、ＨＩＰＡＣＥの標準開発手順を型としてリポジトリに登録し、プロジェクト単位に、カスタマイズして開発作業を誘導する。

リポジトリはシステム開発と保守作業の情報資源倉庫で、支援ツール間での情報の共有、グループワーク支援、関連情報による変更影響分析、アプリケーション単位での再利用等を実現する。リポジトリの情報モデルを用いて、開発作業中の関連仕様情報の検索と参照、仕様変更時の影響を与える相手先への自動通知を行なう。リエンジニアリング支援ツールはリエンジニアリング用に別にデータベースを持ち、既存ソフトウェアを再標準化した標準設計情報をリポジトリに格納する。

<table>
<tr><td>計画技法／
開発方法論</td><td colspan="4">支援ツール</td></tr>
<tr><td rowspan="2">HIPLAN
HIPACE</td><td>ＷＳ型
開発支援ツール
ＳＥＷＢ３</td><td>ホスト型
開発支援ツール
EAGLE2</td><td>第四世代言語
EAGLE/4GL</td><td>リエンジニア
リングツール</td></tr>
<tr><td colspan="4">情報資源倉庫　リポジトリ</td></tr>
</table>

図１　ＣＡＰＳＤＳの体系

３．リエンジアリングの概要

３．１ 背景と目的

リエンジニアリングが登場した背景には、大規模システム開発に伴う保守工数の増加、データベースの統合やダウンサイジング等の環境変化に伴う既存ソフトウェア改造作業の増加や情報システム計画による業務システムの改造作業の増加がある。また、新規開発の場合でも既存のソフトウェア資産を有効に再利用したいとのニーズもリエンジニアリングの背景にある。

保守作業の効率化を図る保守支援ツールは、従来から開発されて来た。しかしこれらの保守支援ツールはフォワードエンジニアリング開発支援システムの一環として開発され、ＣＡＳＥを用いて標準開発した成果物を対象としていた。例えば、ソースプログラムから日本語の仕様書を生成するドキュメントツールは、日本語変換用辞書の整備、プログラムフローの構造化、ルールによる設計情報コメントの記述が前提である。データディクショナリから各種クロスリファレンスを出力するが、格納する関連情報はフォワード開発支援システムが登録するか、システム管理者の手入力で登録していた。

リエンジニアリングでは、ＣＡＳＥ以前の標準化されていないソフトウェアを対象とする。そのため次のような問題点がある。

・コードレベルで変更しており、正確な仕様書が存在しない。
・保守作業が属人的になり、ローテーションが図れない。しかし、既に開発担当者がおらずソースプログラムのみ存在する場合もある。
・変更の繰返しにより、プログラムが複雑になったり、関連ソフトウェア間の不整合が内在化して、変更作業での影響範囲の追跡が困難である。
・古いバージョンの言語仕様、データベース仕様を使っている場合がある。

日立大型オペレーティングシステムＶＯＳ３の主要ユーザを対象にした調査では、４３％のユーザが１Ｍｓｔｅｐ以上のプログラムを保守している結果が出た。大規模データベースシステムを使用しているユーザでは平均約１．４Ｍｓｔｅｐのプログラムを保有している。言語ではＣＯＢＯＬ，ＰＬ／Ｉが主要言語だが、アセンブラ言語も使われている。

リエンジニアリングは、業務システムの保守コスト削減と、既存資産の再利用による開発コスト削減を図る。具体的には次に示す作業を実現する。

①設計情報の抽出

既存ソフトウェアに埋もれている設計情報を抽出し、上流工程での保守作業を実現する。上流工程での保守作業で、ソフトウェアの理解と変更箇所検出の作業工数を短縮する。

②設計情報の標準化

抽出した設計情報を標準化し、変更作業の効率化とプログラム品質の向上化を図る。また、設計情報をリポジトリに移行し、フォワードエンジニアリング開発支援システムを用いて、保守作業の効率化を図ると共に既存資産の再利用に活用する。

③ソフトウェアの構成管理

ソフトウェア構成管理により変更履歴管理、版管理を行なう。また、管理区分や管理基準により論理的な管理情報を付加して資産台帳による資産管理をする。

以上リエンジニアリングの狙いを述べたが、ユーザのニーズは多様で、例えば設計ドキュメントの作成、ソースプログラムの再標準化、フォワードエンジニアリング開発環境への移行、データベースシステムの移行、変更影響分析の機械化等、重点テーマが異なっている。

３．２ リエンジニアリングの体系

「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」がリエンジニアリングである。リエンジニアリングの範囲は、フォワードエンジニアリング環境を含む、ソフトウェアライフサイクルの全工程を範囲にした定義もあるが、本論文では既存ソフトウェアを直接解析、標準化、構造化する技術をリエンジニアリングの範囲とする。図２にリエンジニアリングの概念を示す。

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

リバースエンジニアリングとは、既存ソフトウェアを分析して上流の仕様情報を抽出し、現行仕様の理解を支援する技術である。古くから実用化された技術はソースプログラムからプログラム仕様書を作成するドキュメントツールで、日立ではソースプログラム（COBOL,PL/I）から日本語ＰＡＤ図やＨＩＰＯ図、モジュール関連図等８種のドキュメントを出力するリバースエンジニアリング支援ツール（製品名：ＡＤＣＡＳ）がある。最近では、プログラムの仕様理解、ソースプログラムからの業務ルール抽出、データベース構造の解析と再構築等のリバースエンジニアリング技術がある。

リストラチャリングは主にソースプログラムに対し、保守・再利用が容易な形式に変換する技術で、データ名称標準化やプログラム構造の再構造化等がある。プログラムの仕様やふるまいは変更せず、その表現形式のみを変換する。言語仕様を変換するコンバージョン技術をリストクチャリングに含める場合もあるが、一般的ではない。

リエンジニアリングの方法を図２で説明する。まず、エンドユーザから変更要求を受け付け、リバースエンジニアリング支援ツールで現行仕様の理解と変更影響を検出する。既存ソフトウェアを再標準化する場合は、リバースエンジニアリング支援ツールで抽出した仕様情報と標準化情報から新ソフトウェアに変換する。いずれもリポジトリに格納しフォワードエンジニアリング環境の支援ツールで用いる。

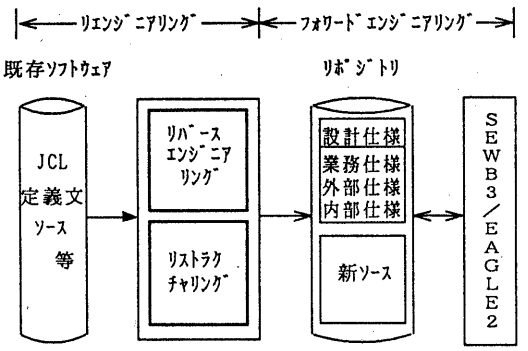

図２　リエンジニアリングの概念

リエンジニアリングの体系に対応して支援体系がある。

①リバースエンジニアリング支援体系

・ジョブフロー逆生成ツール
・プログラム仕様ドキュメント作成支援（対象言語；COBOL,PL/I,C）
・ソース → ＰＡＤ図作成ツール
・リソース関連解析／クロスリファレンス作成支援
・フォワードエンジニアリング開発支援システム用設計仕様作成支援（ジョブフロー仕様、ファイル仕様、レコード仕様、書式仕様、データ項目）

②リストラクチャリング支援体系

・データ名称標準化支援

本論文では、支援ツールのうちジョブフロー逆生成ツールとデータ名称再標準化ツールの機能を解説する。

４．リエンジニアリング支援ツール

リエンジニアリング支援ツールの機能を図３に示す。リソース関連解析はソースプログラム（COBOL）ＪＣＬ（大型オペレーティングシステム　ＶＯＳ３）、ＤＢ定義（ネットワーク型ＤＢ　ＰＤＭII）を解析してプログラム間、ファイル間等の整合性をチェックして関連情報を抽出し、リエンジニアリングＤＢに格納する。ジョブフロー逆生成ツールやデータ項目再標準化ツール等の支援ツールは、このリエンジニアリングＤＢの格納情報を入力してリエンジニアリング作業をする。リエンジニアリングＤＢへのアクセスは、共通のアクセス関数が用意されている。データ名称標準化ツールが扱う同値情報はリソース関連解析で行なう。

リエンジニアリング支援ツールはホスト型支援ツールである。大量の既存ソフトウェアを扱う、ＣＯＢＯＬコンパイラの技術を解析に利用するのが理由である。

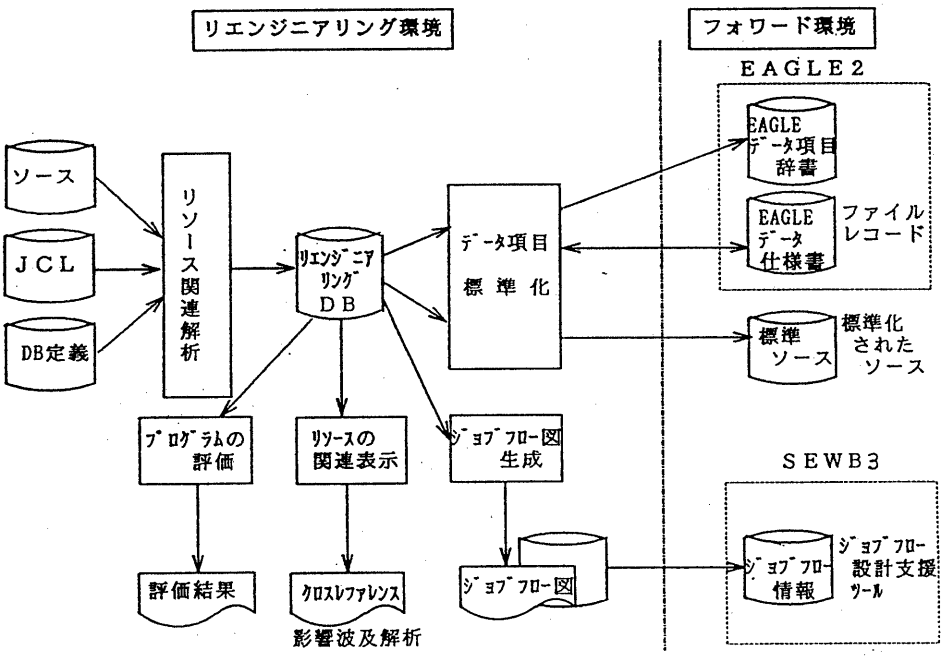

図３．リエンジニアリング支援ツールの概要

４．１　ジョブフロー逆生成ツール

ＪＣＬ、ソースプログラム、ＤＢ定義文を解析してプログラムの実行フローであるジョブフローを生成する。図４にジョブフロー逆生成の流れを示す。

①ＪＣＬ解析情報抽出

ＪＣＬからロードモジュール名とファイル名を抽出する。ＪＣＬだけではファイルの入出力が確定できないのでソースプログラムを解析する。プログラムがユティリティプログラムの場合は、パラメタ解析し入出力情報を確定する。

②ソースプログラム解析情報抽出

ロードモジュール名と同一名称のソースメンバ名を検索する。名称が一致せず検索できない時はロード／ソース対応表をユーザが登録する。また、ソースプログラムが他言語（アセンブラ言語等）の場合や、ソースプログラムが存在しない場合はＪＣＬの物理ファイル名に対応する論理ファイル名と入出力

区分をユーザが設定する機能がある。
ＪＣＬのＤＤ名からプログラム内の論理ファイル名称を抽出し、入出力情報を確定する。ＪＣＬとソースプログラムのファイル情報に矛盾があれば警告メッセージを出力する。例えば、ファイルの数がＪＣＬとソースプログラムで一致しない等の場合で、ソースプログラムの不良なのか、ジョブ運用で該当ファイルがその時不要だったのか支援ツールでは判断できない。

③ジョブフロー図出力

解析情報よりジョブフロー図を作成する(図５)。ジョブフロー仕様はＷＳに転送して、ＳＥＷＢ３ジョブフロー設計支援ツールで編集する。ジョブフロー設計支援ツールは、編集後のシステムフロー図からＪＣＬを生成する。

ジョブフロー逆生成ツールはＪＣＬ解析情報抽出だけでジョブフロー図を出力する機能がある。この場合は、ＪＣＬのＤＤ文のＤＩＳＰオペランドに基づき、推定ルールを用いて入出力区分を判定する。

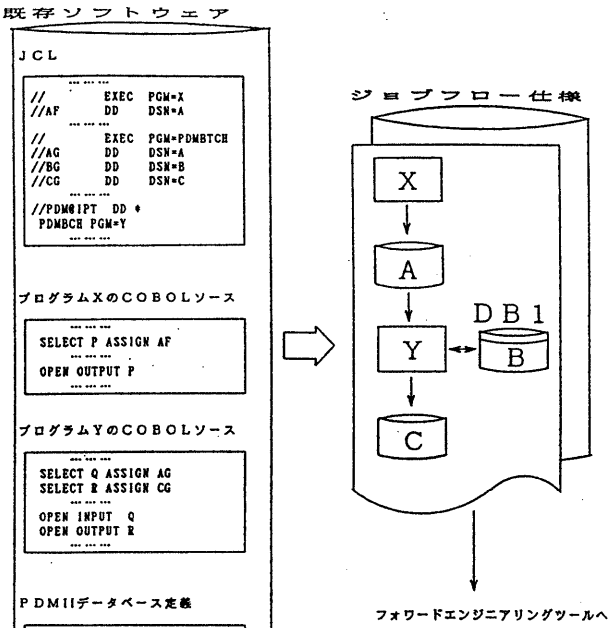

図４　ジョブフロー逆生成の流れ

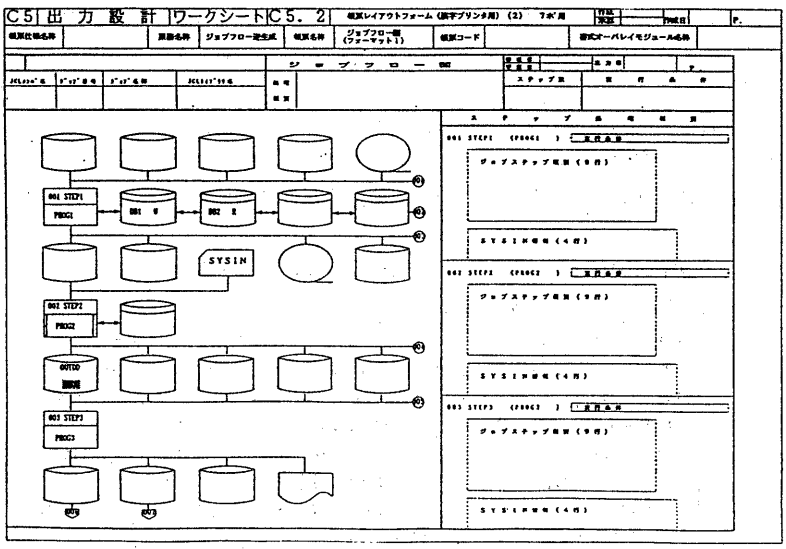

図５　ジョブフロー図

４．２ データ名称標準化ツール

ＣＡＰＳＤＦは、データ中心型開発技法を標準技法にしている。データ中心型技法とは、データを先に正規化／標準化してデータ項目辞書に登録し、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。データ項目辞書でデータ項目を一元管理する。また、データ項目単位のプログラム部品（チェック部品、編集部品）から、ソースプログラムを生成する。

データ名称標準化ツールは、リソース関連解析によりファイル、レコード、データ項目の同値情報を抽出し、名称を統一、標準化する。ソースプログラム内の名称は標準名称に置換する。標準名称はデータ項目辞書に登録する。図６にデータ名称標準化の流れを示す。

プログラムの保守効率を下げる要因のひとつにデータ名称の不統一がある。データ名称の意味理解が困難、データ名称が無秩序に変化している等によりプログラム変更箇所の特定作業の効率を下げている。

不統一のデータ名称には、同音異義語（ホモニム）と異音同義語（シノニム）がある。本支援ツールは、プログラム間の異音同義語（同値データ）を排除する。

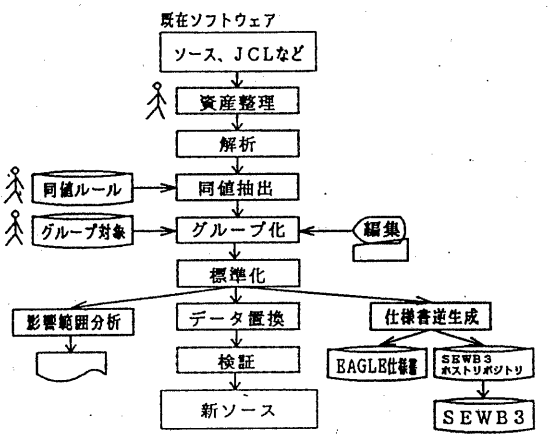

図６　データ名称標準化の流れ

（１） 同値抽出

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。前者の同値データは、ＪＣＬとソースプログラムの解析で検出する。例えばＤＤ名が一致すれば、ファイルは同値関係にあると判定する（図７）。後者の同値データはソースプログラムのデータ間の転送命令等を解析して判定する。ＣＡＬＬ文、ＭＯＶＥ文、ＲＥＡＤ文、ＷＲＩＴＥ文等が解析の対象となる。

（２）グループ化

同値データの情報から下位波及分析を行ない、ファイル、レコード、データ項目をグループ化する。下位波及分析で、例えば同値関係にあるファイルに属するレコードは全て同一グループであると判定する。グループ化の結果をユーザは検証して、データの削除と追加をする。

設定した標準名、標準属性により標準ファイル仕様書、標準レコード仕様書を作成する。ソースプログラム内のデータ名称は、標準名称に置換する。

```
IDENTIFICATION  DIVISION.
 PROGRAM-ID  P1.
ENVIRONMENT  DIVISION.
 INPUT-OUTPUT  SECTION.
  FILE  CONTROL.
   SELECT  ファイル1.
    ASSIGN  TO  FILE01.
DATA  DIVISION.
        :
```

```
IDENTIFICATION  DIVISION.
 PROGRAM-ID  P2.
ENVIRONMENT  DIVISION.
 INPUT-OUTPUT  SECTION.
  FILE  CONTROL.
   SELECT  ファイル2.
    ASSIGN  TO  FILE01.
DATA  DIVISION.
        :
```

－－>ファイル
　の同値情報
（ファイル1とファイル2）
<理由>
ＤＤ名＝FILE01が
同じ。
ただし、P1とP2は、
同一ロードモジュー
ル内のプログラム。

図７　物理的な同値抽出例

５．評価

リエンジニアリング支援ツールの評価目的で、モデルシステムにより手作業による解析作業との比較を行ない効果を測定した。

（１）ジョブフロー生成作業

ジョブフロー生成作業は、対象ジョブの選定と登録から始まり、ＪＣＬとソースプログラムの解析、ジョブフロー図作成、結果の検証までを作業するが、１ジョブ当り約７倍の作業時間短縮効果が出た。ジョブフローの編集とＪＣＬ生成は、ＳＥＷＢ３で機械化されているので、この作業を含めると更に効果は上がる。

ジョブフロー図の確定率は、ＪＣＬだけの場合５１％、ソースプログラム解析を追加した場合は８６％であった。支援ツールでは残り未確定部分は推定ルールにより入出力区分を判断するが、検証結果では問題はなかった。

（２）データ名称標準化作業

手作業に比べて２．５倍の効果があった。モデルシステムのソースプログラムが６本と小規模であり、検証作業が手作業のため効果差が大きく出なかった。しかし、本支援ツールを必要とするユーザは大量プログラムを保有しているユーザであり、効果はもっと大きいと思われる。

６．むすび

本論文では、日立のリエンジニアリング支援ツールを述べたが、リエンジニアリング技術は実用化が始まった段階で、まだ技術的課題は多い。既存ソフトウェア資産から真実を拾い出し、整理する作業は困難が多い。しかし、リエンジニアリングへのユーザの期待は大きく、今後共機能の拡充と適用推進を図って行く所存である。

リエンジニアリング技術は、データ、プロセス、ルールを対象としたリエンジニアリングの他に最近ではビジネスもその対象にする動きもある。この動向にも着目して行きたい。

参　考　文　献

１）原田、他　：ＣＡＳＥの全て、オーム社、ＰＰ．３１５－３２８、１９９１．１１

２）竹下　　：ソフトウェアの保守・再開発と再利用、共立出版、１９９２．９

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シンポジウム、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