

OpenCOBOLの紹介、実績 および長崎県での実例

2012/01/25

OpenCOBOLのご紹介

- 由来

- JMARI(日本医師会総合政策研究機構)のORCA Projectで開発された医療レセプトソフトのために開発された**COBOLコンパイラ処理系**

- 作者：西田圭介 氏

- ORCA Project では2011-12-15現在で**11550**もの医療施設にて利用されている

稼働実績

- 長崎県
 - 元環境 NEC ACOS4(汎用機)
→ OpenCOBOL-1.1
 - 職員総合、財務会計、予算編成 の各業務
 - 約1万5千本、500万ステップのプログラム

稼働実績

- 関西方面某県において県関連団体の会計システム
 - 元環境 N社オフコン
→ OpenCOBOL-1.1
 - 組込SQLを使用しPostgreSQL DBにて運用
→ 80画面、250プログラム 30万ステップ

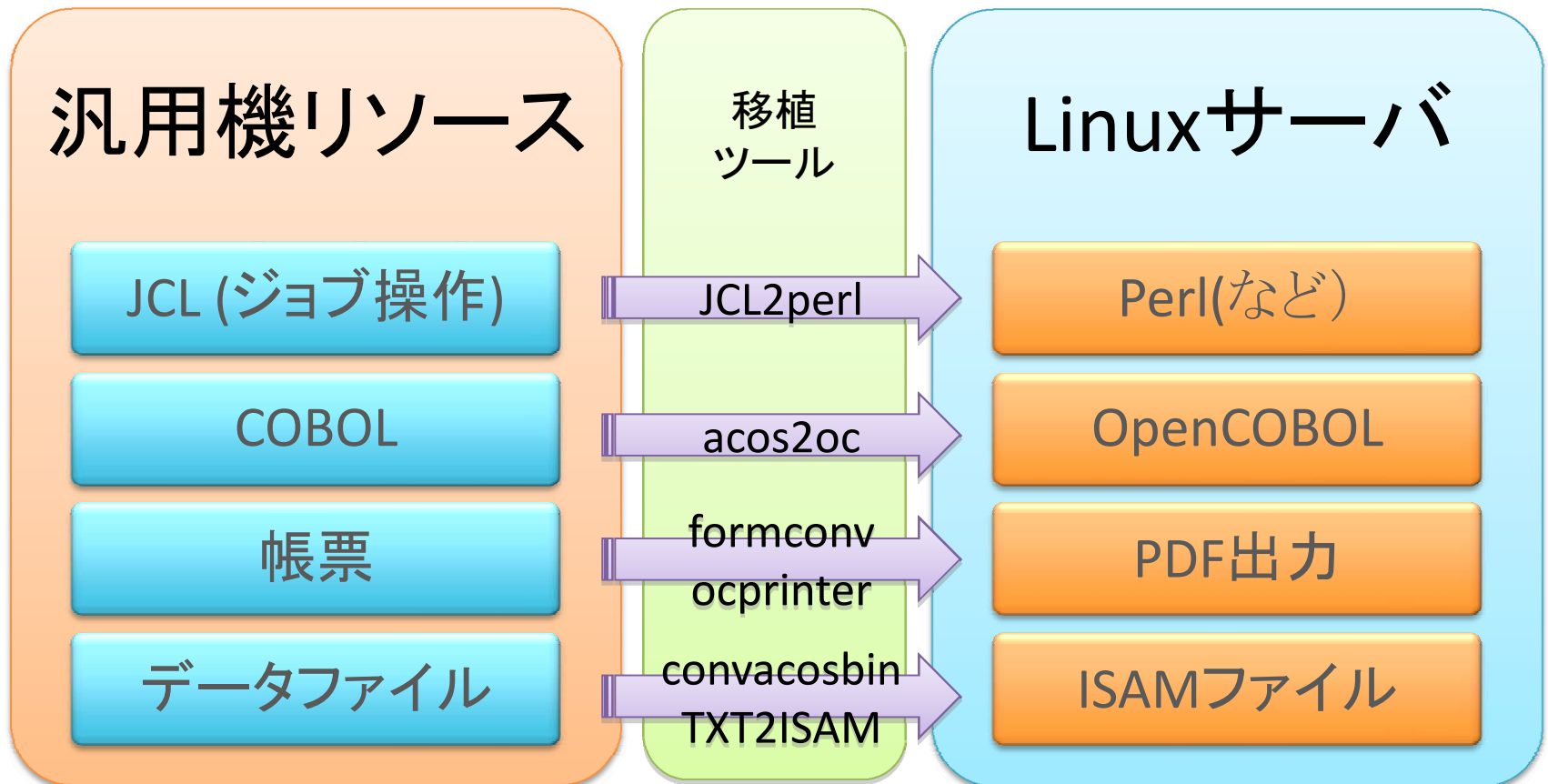
海外での実績

- COBOL-IT 社 (<http://www.cobol-it.com>)
 - 汎用機ダウンサイジング専業
 - 実績
 - フランス財務省(一部)
MicroFocus → OpenCOBOL
 - ロイヤル・アホールド(英国:世界3位の小売業者)
IBM COBOL → OpenCOBOL
 - FlyBuys (オーストラリア:カード、買物ポイント付与)
Fujitsu COBOL → OpenCOBOL

解決すべき問題

- 既存COBOLソースの変換
- 周辺ツールを含めたJCLの移植
- データ変換
- 帳票の移植

移行イメージ



移植ツールについて

- JCL2perl
 - バッチジョブ用JCL言語をPerlスクリプトへ変換するトランスレータ
- acos2oc
 - 汎用機COBOLプログラムをOpenCOBOL用に自動書き換えと依存性を考慮したコンパイルを行う。長崎県庁にて汎用機ダウンサイジング用にカスタマイズしたOpenCOBOLと組み合わせることで90%以上のソースはそのままコンパイル可能。

既存COBOLソースの変換と問題点

- データ型、演算精度の違い
 - COMP-1 浮動小数点、丸め誤差
- 文字の取り扱い
 - 全角文字列、SPACEの解釈
- 文字罫線
 - 外字領域にマップされた文字罫線
- 外字
 - 地名、人名、特殊記号

既存COBOLソースの変換と問題点

- 自動変換と移植工数
 - 業務ロジックを変えるわけではないので、ほとんど機械的に置き換えが出来る
 - 自動変換を検討
 - 自動変換でカバーしていない無いパターンは都度パターン追加
 - 移植ノウハウの自動化

既存COBOLソースの変換と問題点

- 自動変換出来ない箇所
→ OpenCOBOLを機能拡張

1カ所の機能追加でかなりの効果がある
印刷関連などメーカー製COBOL特有の機能

周辺ツールを含めたJCLの移植

- JCL
 - データコピー (CREATESEQ, CREATEVSAS, DD)、並べ替え (SORT)、抽出処理、印刷指示等
- JCLの変換
 - N社系
 - JCL2Perl等で変換
 - I社系 (I社, F社, H社)
 - MVS-JCL言語のインタプリタ実装

JCL移植例 (JCL)

```
¥JOR  OUTID=SPR OUTDEV=PR/M050/JSTD CLASS=A      ;

¥COMM  月次パラメタSTORE;
¥INPUT  ACCEPT1,TYPE=DATA,LIST=YES;
AIDF0270 1 4210918          例月(1)+△+支給日
AIDF0271 4210822 4210918    償還完了者リスト抽出期間(自~至)
*****                    * 前回支給日の翌日~今回支給日
¥ENDINPUT;

¥CREATESEQ
      INFILE=(ACCEPT1 FILESTAT=SYSIN )
      OUTFILE=(AID.F089K RECSIZE=80 BLOCKSZ=11440 RECFORM=FB BPB=4);

¥COMM  引去金ファイル作成;
AIDM2510:¥STEP  AIDM2510 FILE=AIA.IL1 DUMP=DATA;
          ¥ASSIGN AIDF0010 AID.F101  SHARE=ALL HOLDMODE=NO;
```

JCL移植例 (Perl)

```
#!/usr/bin/perl
use strict;
use warnings;
use JCL;

JOR "OUTID=SPR OUTDEV=PR/M050/JSTD CLASS=A";

COMM "月次パラメタSTORE";
INPUT "ACCEPT1,TYPE=DATA,LIST=YES", <<_EOT;
AIDF0270 1 4210918 例月(1)+△+支給日
AIDF0271 4210822 4210918 償還完了者リスト抽出期間(自~至)
***** * 前回支給日の翌日~今回支給日
_EOT
ENDINPUT;
CREATESEQ
    "INFILE=(ACCEPT1 FILESTAT=SYSIN)",
    "OUTFILE=(AID.F089K RECSIZE=80 BLOCKSZ=11440 RECFORM=FB BPB=4)";

COMM "引去金ファイル作成";
AIDM2510: STEP "AIDM2510 FILE=AIA.IL1 DUMP=DATA";
            ASSIGN "AIDF0010 AID.F101 SHARE=ALL HOLDMODE=NO";
```

周辺ツールを含めたJCLの移植

- ソートパラメータ等の処理について
 - 並べ替えや抽出について細かい指示、
 - COBOLプログラム等で動的に生成している場合がある
 - 変換せずにそのまま使用

周辺ツールを含めたJCLの移植

- 周辺ツールの代替方法
 - 並べ替え : Linuxのファイルソートを外部使用
 - サーバ内蔵メモリを有効活用し実機の数倍高速
 - 抽出 : 複雑なルール適用が必要なため perl等COBOL以外の言語で実装
 - レコード件数分処理が必要であるが、並列化など高速化の余地はある
 - 印刷 : 印刷用の外部プログラム

並行運用での動作検証

- 既存業務、他業務とのデータ連携
 - DBと汎用機系とのデータ相互変換
- データ突き合わせ
 - 移行元と同じ入力データから同じ出力データが得られるか？
- 処理速度
 - バッチが予定時刻までに処理完了するか？

帳票資産の活用

- COBOLプログラムと連動した帳票の移行
 - 改行、改ページ等の処理についてのルールがCOBOLプログラムに埋まっている
 - 例：連続した同じデータは1ページに纏める等
→ **なるべくそのまま移植したい**
 - 帳票レイアウトデータは罫線位置などデザインの情報が記載されている
 - 線の開始位置、長さ、太さ等

帳票資産の活用

- 汎用機プリンタエミュレーションの利点
 - プログラム内のアルゴリズムをそのまま活用
→ 改行、改ページのロジック等
 - 帳票レイアウトの再デザインが不要
→ 現在の運用を変えないまま移行可能

移植作業の実際

- 汎用機COBOLから、OpenCOBOLへの移植
 - 自動化しているためコンパイルを通す段階まではほぼノータイム
(例：200ソースの変換とコンパイルまで2時間)
- JCLの移植
 - 自動化しているため移植自体の工数は少ない
 - 動作確認用データの準備に時間がかかる

移植作業の実際

- データ移行
 - レイアウトを考慮しながらの変換作業
 - FILLER等の未使用と思われた箇所にデータが埋められている場合がある
 - 基本的に手作業
- 帳票の移植
 - メーカー製の帳票マクロプロセッサの機能をどれだけ使っているか？

移植作業の実際

- 移植作業工数
 - COBOL 5%
 - JCL 5%
 - データ 50%
 - 帳票 40%
- 検証作業に一番時間が必要