

COBOL記述APのMIA移行性の
一考察

4K-3

奥山 晃、水嶋 宏也
NTT情報システム本部

1. はじめに

業務APの開発規模は年々膨れ上がり膨大なものとなっており既存業務APの再利用技術が重大な課題となっている。マルチベンダ環境間でのAPの流通性を確保するためにNTTの調達仕様としてAPのプラットフォームとなるMIA仕様を制定した。COBOL言語仕様は移行性の高い言語であるが、拡張COBOL仕様を有するDIPS COBOLで記述された業務APから基本的なCOBOL仕様を持つMIA COBOL仕様への移行性について評価再確認したので報告する。

2. 背景

近年のコンピュータの普及にともない、コンピュータ・システムがカバーすべき業務分野が多岐にわたるようになってきた。複数のベンダのコンピュータを組み合わせ、個々の業務分野にこれを最も得意とするコンピュータを適用すれば、広範な業務分野を総合的にカバーするシステムを構築することが期待できる。このような複数のベンダのコンピュータによって構築されたシステムは「マルチベンダ・システム」と呼ばれる。しかしながら、1) コンピュータ間でのアプリケーション・プログラムのポータビリティが低い、2) コンピュータ同士がうまくつながらない、あるいはつながったとしても効率的な協調処理が難しい、3) コンピュータ間で使い勝手が異なる等の問題がある。そこで、マルチベンダ・システム環境においてベンダ固有のハード・ウェアやソフトウェアの特徴をいかしつつ、異なるベンダ間でのAP開発の一元化、APのポータビリティと運用性の確保を目的としてNTT及びNTT DATAは公募により得たパートナー国内外の5社との共同研究によりMIA (Multivendor Integration Architecture) 第1版を1991年1月に制定した。

従来、NTTではDIPSを中心とした業務処理システムを開発してきており、膨大な既存APを有している。今後、MIA仕様を適用したシステムに効率的に移行するには、既存業務について開発済の業務APを再利用する必要性があり、DIPS-C

OBOLからMIA-COBOLへの移行性について評価した。

3. MIA COBOL仕様の概要

基本的にはJIS X 3002の必須機能単位の上位水準に従う。しかし、廃要素のように、代替手段があり将来国際標準からも削除される予定の機能や、利用者要求の無い機能は削除し、利用者要求があり必要とされる機能は追加した。また、プログラムの流通性を高めるため、作成者規定項目となっていた部分を可能な限り規定した。

(1) 追加した規定

- ・日本語機能

(2) ベンダ依存項目に対する具体的規定

- ・B領域の長さ
- ・計算機名の規則
- ・英数字編集項目の文字位置の数

(3) 削除した規定

- ・実際に使用されていない機能の削除：通信機能単位、報告書機能単位、外部スイッチ、不定ファイル
- ・旧標準との互換のために残されている機能(廃要素)：区分化機能単位、デバッグ機能単位

4. 移行の阻害要因

以下にMIA-COBOL仕様への移行に対して阻害要因となる主な項目を述べる。

- (1) HEX コード表現。
- (2) BASED属性。
- (3) POINTER属性。
- (4) BIT属性。
- (5) ADDRESS関数。
- (6) EXAMINE文。
- (7) TRANSFORM文。

5. 移行阻害要因のチェック方法

我々はDIPS-COBOLからMIA-COBOLへの移行を支援するためにMIA用COBOLチェッカをすでに開発した。本評価では、このチェッカにより確認する方法を採った。MIA用COBOLチェッカの概念を図1に、チェッカによるチェック結果例を図2に示す。

6. 評価方法

- (1) トランザクション制御部分は移行対象外とし純業務処理部分について評価する。
- (2) 評価対象プログラムはNTTの社内システムで稼働中の業務処理プログラムとする。
- (3) オンライン・プログラム、バッチ・プログラム、共通・プログラムに分類し評価する。
- (4) それぞれの分類から無作為に評価候補を抽出する。
- (5) 評価はライン数に対する比率でおこなう。
- (6) 単純に変換出来る部分、単純外部情報により変換できる部分、人手で修正を要する部分に分けて評価する。

7. 評価結果

評価結果を表1、表2、表3に示す。

8. 考察

- (1) 修正すべき量は、オンライン・プログラム、バッチプログラム、共通プログラムで特に差はなく、約10%である。
- (2) 人手で修正を要する部分は、共通・プログラ

ムには存在せず、オンライン・プログラム、バッチ・プログラムで全修正量の約10%である。その内約半分は、多種の異なる属性間で再定義された構造体である。

9. 今後の課題

本報告は、あるシステムの業務APからサンプリングして調査した。今後システム数および、サンプルAPを拡大し精度を上げる予定である。

10. おわりに

移行性の高い言語であると言われているCOBOL言語についてDIPS-COBOLからMIA-COBOLへの移行性という課題で評価したが、機械的変換可能部分が修正量の約90%に達しており移行性の高い言語であることを再確認した。さらに本結果は移植ツールの実現性を示している。また、MIA-COBOL仕様では3項で述べたように移行性を重視した仕様となっているので、業務APの流通性の向上が期待できる。

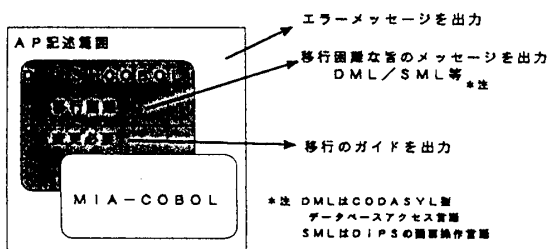


図1. MIA用COBOLチェッカの概念

```
IDENTIFICATION DIVISION.
PROGRAM-ID.      W8XXXX.
AUTHOR.         OKUYAMA.
/* 000 (D) AUTHORはDIPS固有の文です。削除して下さい。 */
03 DW-AR1RKSU-1 PIC 9(2) COMP.
/* 000 (W) USAGE COMP句はDIPS仕様とMIA仕様とで扱いが異なるので注意が
   必要です。 */
TRANSFORM DW-AR1RDHJ-G (D104) FROM " " TO "0"
/* 000 (E) TRANSFORM文はDIPS固有の文です。INSPECT文に書き換えて下さい。
/* 000 (E) DML文はすべて書き換えて下さい。 */
```

*注1 はチェッカによる警告です。 *注2 MIFESと混同した例です。

図2. チェッカによるチェック結果例

表1. オンライン・プログラム

項目	CASE	CASE1	CASE2	CASE3	CASE4	CASE5	合計
ライン数		6498	209	823	1418	2419	11367
修正すべき量		9.70%	10.05%	10.33%	15.87%	8.93%	10.35%
単純変換		5.14%	9.57%	7.29%	6.00%	5.13%	5.48%
単純外部情報		4.08%	0.00%	2.79%	7.26%	2.65%	4.00%
人手による修正		0.48%	0.48%	0.24%	2.61%	1.16%	0.87%

表2. バッチ・プログラム

項目	CASE	CASE1	CASE2	CASE3	CASE4	CASE5	合計
ライン数		567	915	1636	4658	4239	12017
修正すべき量		7.58%	10.16%	4.70%	12.84%	9.62%	10.14%
単純変換		6.70%	7.87%	3.72%	10.63%	7.76%	8.28%
単純外部情報		0.53%	2.19%	0.49%	1.18%	0.87%	1.02%
人手による修正		0.35%	0.11%	0.49%	1.03%	0.99%	0.84%

表3. 共通・プログラム

項目	CASE	CASE1	CASE2	CASE3	CASE4	CASE5	合計
ライン数		792	159	428	542	492	2413
修正すべき量		19.07%	6.92%	4.21%	5.54%	7.72%	10.28%
単純変換		15.91%	8.33%	2.46%	5.35%	7.32%	9.08%
単純外部情報		3.16%	0.00%	0.23%	0.18%	0.41%	3.97%
人手による修正		0.00%	0.00%	0.00%	0.00%	0.00%	0.00%

参考文献

- (1) Multivendor Intergration Architectur, Version 1.1 第1編概説 テクニカル・リクワイメント Tr55000-1, NTT, Apr., 1992.
- (2) Multivendor Intergration Architectur, Version 1.1 第2編アプリケーションプログラム・インタフェース仕様 プログラミング言語COBOL、Tr55000-1, NTT, Apr., 1992.