ＭＩＡ仕様を用いたＡＰの移植性についての一考察

4K－4

山田　竜二*　瀬尾　紳一郎*　塩野入　理**
（＊ＮＴＴ　情報システム本部　＊＊ＮＴＴ　情報通信網研究所）

１．はじめに

複数ベンダ環境の計算機を組み合わせてシステムを構築する場合、ＡＰの移植性等が問題となる。ＮＴＴでは、この問題を解決し複数ベンダによる競争のもとでより良い製品とサービスを得るために、小型から大型計算機までを対象としたＭＩＡ（Multi-vendor Integration Architecture）仕様を規定し、ＭＩＡによるシステム構築を推進している。本稿では、ＭＩＡ仕様で記述されたＡＰの移植を実際に行ったので、その移植性について報告する。

２．ＭＩＡ仕様の移植性とベンダ依存性

ＭＩＡでは国際標準言語（ＣＯＢＯＬ、Ｃ、ＳＱＬ等）をもとに必要に応じて仕様の追加／削除を行い、さらにトランザクション処理言語としてＳＴＤＬ（Structured Transaction Definition Language）を新たに採用し、ＡＰＩ（Application Program Interface）を規定している。更に、国際標準言語で実装固有とされている項目について可能な限り規定している。従って、ＡＰの高い移植性が保証されるようになっている。しかし、一部にベンダの特徴を生かすために、実装に自由度を持たせる等、ベンダ毎に実装固有な項目が残っている。

１）言語仕様自体の総ベンダ固有項目数
- ＳＴＤＬ・・・２５
- ＣＯＢＯＬ・・・３４
- Ｃ・・・１１８
- ＳＱＬ・・・３４
- ＦＯＲＴＲＡＮ・・・３５

２）実装に自由度を持たせるための総ベンダ固有項目数
- ＳＴＤＬ・・・４７
- ＣＯＢＯＬ・・・０
- Ｃ・・・０
- ＳＱＬ・・・０
- ＦＯＲＴＲＡＮ・・・０

３．移植結果

今回移植を前提に開発を行ったＡＰ（共通モジュール）は、以下の特徴を持つ。

（１）ＮＴＴのＭＩＡ導入全システムで共通的に使用されるプログラムである。（図１参照）
- 業務ＡＰ制御機能：システム個別のＡＰ起動依頼を受信して起動可否をチェック、起動すべきＡＰを選択起動する機能
- オンラインサービス管理機能：共通モジュールの開始／終了処理を矛盾なく実行する機能
- コンソール入出力機能：コンソールコマンドの受信及びコンソールにメッセージの出力を行う機能
- スケジュール管理機能：あらかじめ設定されたスケジュールに従って、ＡＰの起動／ファイル転送を行う機能
- ファイル転送管理機能：ファイル転送を矛盾なく行うために転送状態／競合制御の管理、再転送処理を行う機能

（２）ＳＴＤＬ言語とＣＯＢＯＬ言語で記述

我々は、ベンダ実装固有項目が存在する中で、ＡＰの移植性を向上させるために次のような対策を講じた。

①ＭＩＡが規定している（各ベンダがサポートしなければならない）最小アーキテクチャ定数の範囲内で開発

②ＭＩＡで実装固有としている言語部分は極力使用せず、ポータブルな言語の範囲で開発

On portability of application program coded by MIA specification

Ryuji Yamada* Shin-ichiro Seo* Osamu Shionoiri**
*NTT Information Systems Headquaters **NTT Network Information Systems Laboratories

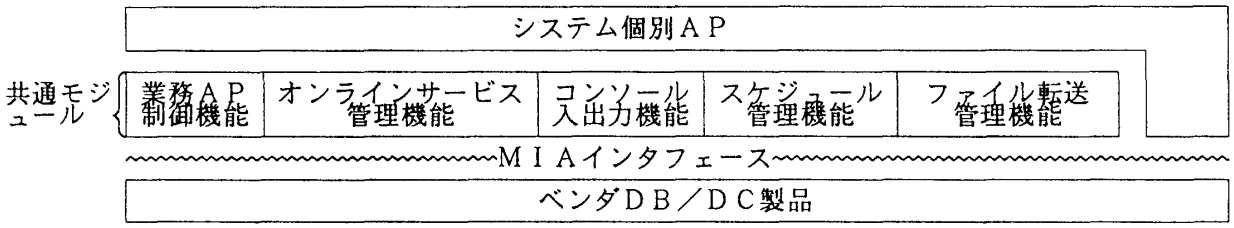

図１　共通ＡＰの位置付け

③例外の扱い等、ベンダが実装時の都合により、ベンダが制限可能な項目を加味して開発

④ＭＩＡで共通的に規定しきれなかった機能に対しては、一部分に局所化しベンダ依存部として隠蔽。

⑤仕様が正しく反映されず、ベンダ固有の記述がされている場合移植後、正常に動作しない可能性がある。この様な問題を回避するために、開発段階でツールによるＡＰのＭＩＡ仕様言語チェックを実行。

⑥重複する可能性のある各種資源名称の付与基準を規定し、システム個別のＡＰを含めて今後のＡＰ開発においてもこの規定が反映されるものとした。

今回は、異なる３ベンダ（Ａ、Ｂ、Ｃ）環境間での移植性の検証を行った。表１に示すように、高い移植性を示した。ただし、ＭＩＡ規定外の機能を使用しなければ実現できない機能（ベンダ依存部）については移植対象外とした。

表１　共通ＡＰ移植結果

移植方向	言語	移植規模	修正項目数	修正規模	移植率
A→B	STDL	31Kｽﾃｯﾌﾟ	0	0ｽﾃｯﾌﾟ	100%
	COBOL	56Kｽﾃｯﾌﾟ	4	0.7Kｽﾃｯﾌﾟ	98.8%
B→A	STDL	19Kｽﾃｯﾌﾟ	0	0ｽﾃｯﾌﾟ	100%
	COBOL	44Kｽﾃｯﾌﾟ	4	2Kｽﾃｯﾌﾟ	95.5%
B→C	STDL	19Kｽﾃｯﾌﾟ	0	0ｽﾃｯﾌﾟ	100%
	COBOL	20Kｽﾃｯﾌﾟ	1	0.8Kｽﾃｯﾌﾟ	96.0%
平均	STDL				100%
	COBOL				97.1%

4．考察

オンラインＡＰの開発には実装固有項目の全てを必要としているわけではない。少なくとも共通ＡＰが必要とした言語レベルでの項目は以下に示すもののみである。

・COBOLのASSIGN句・・・１％未満
・COBOLのCOPY句・・・約３％
・COBOLプログラムの最後尾にEND PROGRAMの有無・・・１％未満
・ﾌｧｲﾙｽﾃｰﾀｽ・・・１％未満

ＣＯＢＯＬに関する項目は、ほとんど全てのオンラインＡＰで必須となる項目であるのでＭＩＡ仕様で規定し、ベンダの実装固有項目としない方法もあると思われる。また、これらは機械的に修正可能なものばかりであった。従って、ツール等を作成すれば簡単に移植が可能である。

5．まとめと今後の課題

３で述べたような対策を行えば高い移植性を持つオンラインＡＰが開発可能であることが判った。ただし、今回はＳＴＤＬとＣＯＢＯＬのコンパイルまでの移植性の検証しか行っていない。従って以下に示す事項について今後の課題として実施する予定である。

・各ベンダのアーキテクチャの違いに起因して、移植先環境で性能問題が発生する可能性があるため、性能測定までを含めた移植性の検証
・ＳＱＬ言語、Ｃ言語の移植性の検証

参考文献

（１）Mutltivendor Integration Architecture, Version 1.1, 第１編概説 ﾃｸﾆｶﾙ･ﾘｸﾜｲﾔﾒﾝﾄ、TR550001-1, NTT, Apr., 1992.

（２）Mutltivendor Integration Architecture, Version 1.1, 第７編ﾕｰｻﾞ･ｶﾞｲﾄﾞ ｱﾌﾟﾘｹｰｼｮﾝ･ﾌﾟﾛｸﾞﾗﾑ･ﾎﾟｰﾀﾋﾞﾘﾃｨ･ｶﾞｲﾄﾞ、TR550001-1, NTT, Apr., 1992.